Top-*N*协同过滤推荐技术研究

赵向宇

2014 年 6 月

中图分类号：TP391
UDC 分类号：004.8

Top-*N*协同过滤推荐技术研究

作 者 姓 名　　赵向宇
学 院 名 称　　计算机
指 导 教 师　　牛振东教授
答辩委员会主席　林守勋研究员
申 请 学 位　　工学博士
学 科 专 业　　计算机软件与理论
学位授予单位　　北京理工大学
论文答辩日期　　2014 年 6 月

Research on Top-*N* Recommendation with Collaborative Filtering

Candidate Name： Xiangyu Zhao
School or Department: Computer Science and Technology
Faculty Mentor: Prof. Zhendong Niu
Chair, Thesis Committee： Prof. Shouxun Lin

Degree Applied: Doctor of Engineering
Major： Computer Software and Theory
Degree by: Beijing Institute of Technology
The Date of Defence： June，2014

Top-N协同过滤推荐技术研究

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签　　名：　　日期：

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签　　名：　　日期：

导师签名：　　日期：

摘要

随着计算机和网络技术的飞速发展，互联网逐渐走进了人们的日常生活，彻底改变了人们获取信息的方式。海量的网络信息为满足用户的信息需求提供了保证，给人们带来了极大的便利。但是，网络信息的多样性和多变性导致了信息的过度膨胀，带来了信息过载问题，使人们难以快速、准确地从浩瀚的信息资源中寻找所需要的信息。推荐系统通过对用户行为数据进行分析、建模，预测并推荐用户可能感兴趣的产品，可以一定程度上解决信息过载问题。另一方面，推荐系统是大数据背景下研究和应用海量数据的热点领域，是网络时代向信息时代转变的重要技术。正是由于其巨大的理论及应用价值，推荐系统及相关技术成为近年来研究的热门课题，受到了学术界和产业界的广泛重视。

推荐系统的目标是帮助用户选择一些用户自己可能感兴趣的产品，并将其以合适的形式展现给用户，Top-*N* 推荐是其中的重要问题。协同过滤推荐算法是推荐系统中的关键技术之一，在理论研究和实际应用两方面都获得了长足的发展。但是，随着用户数量和系统规模的不断扩大，协同过滤推荐技术仍然存在着一些亟待解决的问题，包括数据稀疏性、冷启动、流行偏置、可扩展性、动态性、准确率和多样性等。本论文主要研究解决其中的数据稀疏性和流行偏置问题，具体研究内容和创新成果包括：

1) 针对协同过滤推荐中的流行偏置问题，即传统推荐算法往往倾向于推荐流行度较高的产品，提出了一种基于意见的协同过滤推荐算法。本论文从用户行为数据的产生过程对用户模型的影响进行分析，考虑用户行为受到各种广告、口碑、推荐等因素的影响，针对不同行为对用户模型的表达能力不同，利用用户意见信息和产品流行度信息构建用户行为的置信度函数，并使用该置信度函数调节产品对用户模型的影响，提出了基于意见的协同过滤推荐算法。实验结果表明该算法比传统的推荐算法具有更好的 Top-*N* 推荐准确率和多样性，可以有效缓解推荐系统的流行偏置问题。

2) 针对协同过滤推荐的数据稀疏性问题，提出对缺失数据中的负例信息建模的方法，并将其用于改进矩阵分解推荐算法。数据稀疏性问题是指推荐系统中的用户-产品评分矩阵极其稀疏，大量评分数据缺失，为推荐系统挖掘用户兴趣、向用户推荐产品带来了极大挑战。本论文考虑在数据稀疏背景下，缺失数据中包含用户兴趣的负例信息，分别提出加权法、随机抽样法和近邻抽样法三种对缺失数据建模的方法，识

别其中用户兴趣的负例信息，并利用这些负例信息调节推荐模型的训练过程，改进矩阵分解推荐算法。实验结果表明这些改进算法有效提升了基线算法的 Top-*N* 推荐准确率和多样性。

3) 针对协同过滤推荐算法中传统用户行为模式假设用户随机选择产品并评分的局限性，考虑用户选择产品并决定对其评分本身就是一种用户兴趣的体现，提出两阶段用户行为模式和两步预测推荐算法。两阶段用户行为模式将用户选择产品进行评分和用户给出对产品的评分值区分开来，将其视为用户行为的两个阶段。为验证两阶段用户行为模式的有效性，本论文分析了它与传统用户行为模式的区别，并利用真实的推荐系统数据集进行了数据分析和验证。之后，通过对两阶段用户行为模式的仿真，提出一种两步预测推荐算法框架，分两步预测用户对产品评分的概率和用户对产品的评分值，然后整合两步预测的结果完成推荐任务。最后，本论文提出了两种两步预测推荐算法框架的具体实现，分别是基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。实验结果表明，两步预测推荐算法的 Top-*N* 推荐效果优于主流推荐算法，验证了两阶段用户行为模式和两步预测推荐算法的有效性。

关键词：Top-*N* 推荐；协同过滤；流行偏置；数据稀疏性；缺失数据建模；两阶段用户行为模式；两步预测推荐算法

Abstract

With the rapid development of computer and web technology, the Internet runs gradually into people's daily lives, and a complete change has occurred in the way for people accessing information. Vast amounts of web information provide lots of useful contents. However, this lead to a problem named inforamtion overload, which makes it becoming increasingly hard for people to find relevant information. Recommender systems have been introduced in recent years to help people in retrieving potentially useful information or products. Meanwhile, recommender system is a hot research topic in the field of Big Data, and an important transition technology from the web age to the information age. Since the great value of recommender system in theory and application, it gains much attention from both academia and industry.

The main goal of recommender systems is to find some recommendations that match users' interests. In hence Top-*N* recommendation is the core target. Collaborative filtering (CF) is a leading approach to build recommender systems which has gained considerable development and popularity. However, there are still some drawbacks which hinder its further development, including data sparse, cold start, popularity bias, scalability, temporal, accuracy and diversity. This paper focuses on solving data sparse and popularity bias of CF, in order to gain improvement on Top-*N* recommendation task. The main research works and contributions are listed as follows:

1) Existing recommender systems using CF suffer from popularity bias problem. Popular items are always recommended to users regardless whether they are related to users' preferences. In order to solve it, this paper proposes an opinion-based CF approach (OWUserCF). By analyzing the reasons and influences of popularity bias and comparison with some solutions, an assumption is proposed that a user's rating on an item means differently to his/her preference according to the popularity of the target item. Based on the assumption, OWUserCF introduces weighting functions to adjust the influences of popular items according to item popularities and user opinions. Experiment results show that OWUserCF outperforms the baseline approaches on Top-*N* recommendation task, which indicates that OWUserCF has solved the popularity bias problem in a certain degree.

2) Data sparse is a main challenge in the area of CF. That is there are only a few observed ratings, lots of items which have not been rated are missing data. This paper

focuses on finding the reason why these data are missing. It is found that data are missing not at random. A part of missing data is due to that users choose not to rate them. This part of missing data are negative examples of user preferences. Utilizing this information is expected to leverage the performance of recommendation algorithms. Unfortunately, negative examples are mixed with unlabeled positive examples in missing data, and it is hard to distinguish them. This paper proposes three schemes to model the negative examples in missing data, including weighting scheme, random sampling scheme, and neighbor-based sampling scheme. The schemes are then adapted with SVD++, which is a state-of-the-art matrix factorization recommendation approach, to generate recommendations. Experiment results show that our proposed approaches gain better Top-*N* performance than the baseline ones on both accuracy and diversity.

3) Conventional CF approaches are based on an implicit underlying assumption that users randomly select the items which they rate in recommender systems. However, users are free to choose which items to rate. In our opinion, users always rate the items that they want to rate, especially in the age of information overload. As a result, a Two-layer Model of User Behavior (TMUB) is proposed by dividing user behaviors into two layers. The first layer is that the current user selects an item to rate. The second one is rating it with a value. This paper analyzes the difference between TMUB and conventional model, and verifies the effectiveness of TMUB by data analysis with a realworld dataset of recommender systems. A two-step recommendation framework is then proposed as a simulation of TMUB. That is predicting the probability that user *u* rates item *i* (in the first step), and then predicting the value which *u* may rate *i* with (in the second step). Furthermore, two-step neighbor-based recommendation algorithms and a two-step model-based recommendation algorithms are proposed based on the framework. Experiment results show that these two-step recommendation algorithms outperform benchmark approaches on both accuracy and diversity, which demonstrate the effectiveness of TMUB and two-step recommendation algorithms.

Key Words: Top-*N* recommendation; collaborative filtering; popularity bias; data sparse; modeling missing data; two-layer model of user behavior; two-step recommendation algorithm

图索引

表索引

目录

第1章 绪论

1.1 本论文研究的背景和意义

随着计算机和网络技术的飞速发展，互联网逐渐走进了人们的日常生活，彻底改变了人们获取信息的方式。在互联网这个具有开放性、动态性和异构性的全球分布式网络中，蕴含着海量的具有巨大价值的文档、图像、视频等信息，并且每天都在不断的更新和变化。据美国因特网检测公司 Netcraft 的月度监测报告[1]指出，截止 2013 年 12 月，全球互联网站数量超过 861,023,217，图 1.1 是该报告中展示的互联网网站数量变化曲线。

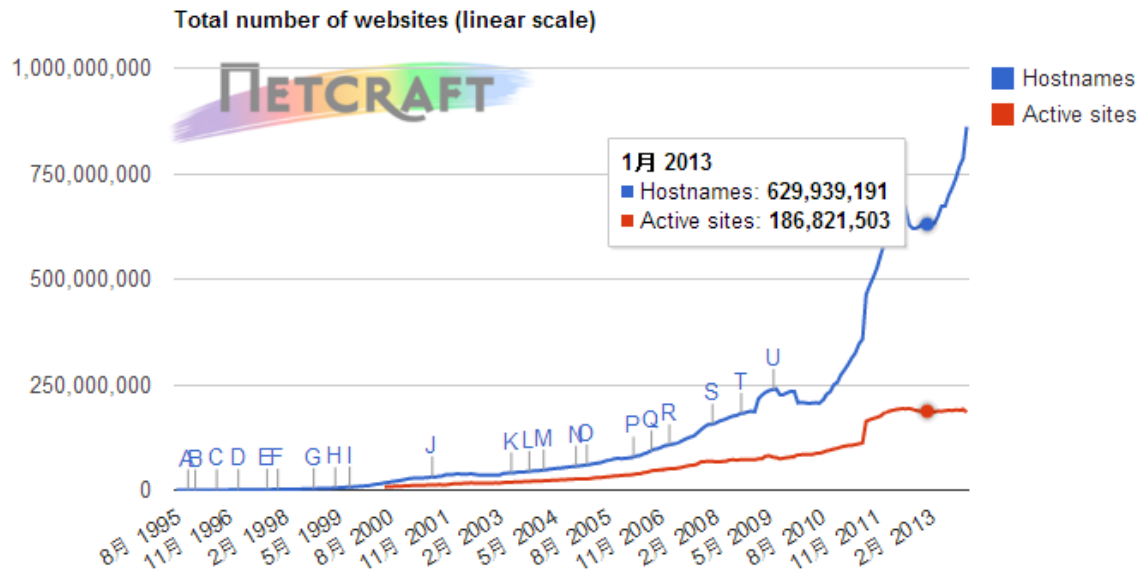

图 1.1 互联网网站数量变化图

从图 1.1 中可以看出，网站数目在飞速的增长，仅 2013 年一年网站数量就从 6.3 亿增长到了 8.6 亿，增长幅度达到了 37%。海量的网络信息为满足用户多种多样的信息需求提供了保证，给人们带来了极大的便利。但是，网络信息的多样性和多变性导致了信息的过度膨胀，带来了信息过载问题（Information Overload[1]），使人们难以快速、准确地从浩瀚的信息资源中寻找所需的信息。为解决信息过载问题，互联网技术经历了导航、检索和推荐三个主要发展阶段[2, 3]：

[1] http://news.netcraft.com/archives/category/web-server-survey/

1) 导航

导航技术主要应用于两种类型的网站中。一种是以 Yahoo、新浪、搜狐等为代表的门户网站，它们基于分类目录体系，对互联网中的信息进行导航，以便用户分门别类进行信息浏览。另一种是以 DMOZ、Hao123 等为代表的分类目录网站，它们将著名的网站分门别类，以便用户根据类别查找相应网站[4]。图 1.2 列举了几个著名的导航类网站。

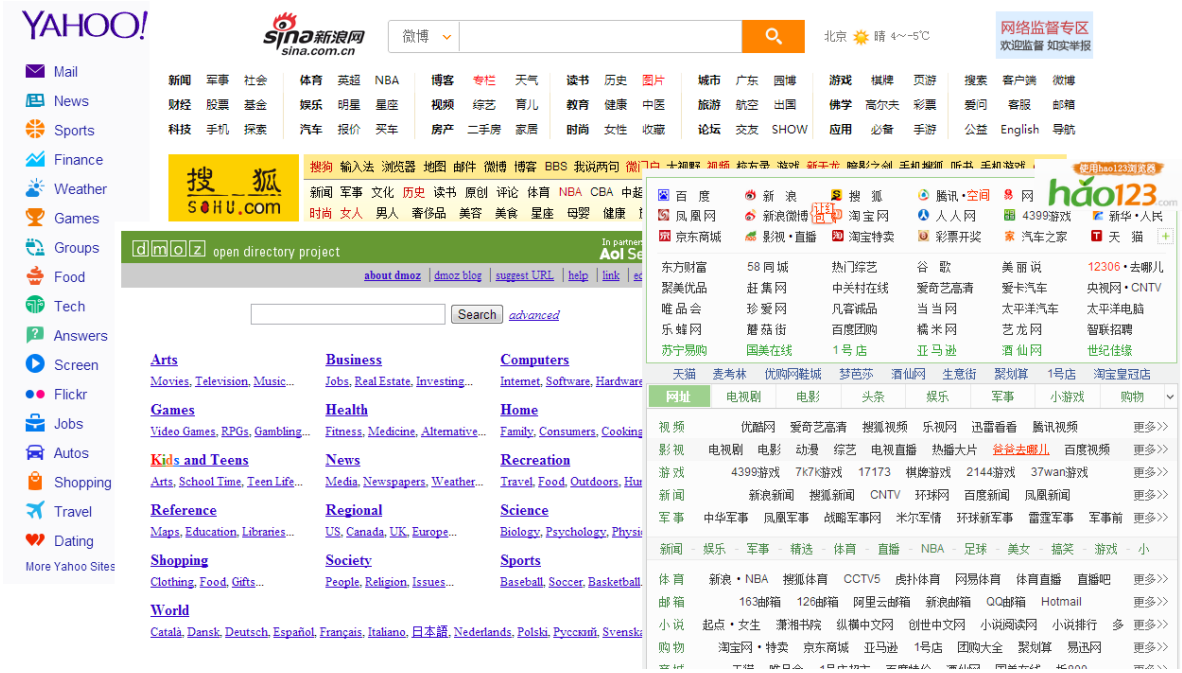

图 1.2 著名导航类网站举例

2) 检索

随着网络信息的逐渐增多，导航网站已无法帮助用户定位网络上大量涌现的信息。这时，以谷歌、百度等为代表的基于信息检索技术的检索系统发展壮大起来，成为了人们获取信息的主要方式。检索系统根据用户输入的关键词，返回给用户与关键词相关的网页。与导航网站不同的是，检索系统不只是简单地返回相关网页，还依据网页与关键词的相关性以及网页本身的重要性对网页进行排序，使得用户需要的网页尽可能的展现在结果的前面。图 1.3 列举了几个著名的检索系统。

图 1.3 著名检索系统举例

3) 推荐

信息检索的方法为人们解决信息过载提供了很大的帮助，但也存在着一些无法解决的问题：检索往往通过关键词的匹配帮助人们进行信息的过滤，但是人们的信息需求有时很难用几个关键词简单表示；传统的检索方法只能呈现给所有用户一样的检索结果，无法针对不同用户的兴趣爱好提供相应的服务；另外，人们有时并不明确知道自己的需求，这样就更难使用检索系统获取信息了。推荐系统可以一定程度上解决以上问题。推荐系统不需要用户提供明确的需求，而是通过分析用户历史行为建立用户兴趣模型，进而向用户推荐符合其兴趣和需求的信息。因此，从某种意义上说，推荐系统和检索系统对用户来说是两个互补的工具。检索系统满足了用户有明确目的时主动查找的需求，而推荐系统能够在用户没有明确目的的时候帮助他们发现让他们感兴趣的新内容[2]。

自 20 世纪 90 年代以来，推荐系统的研究已有了长足的发展[5]。在学术界，推荐系统已经成为了一个专门的研究领域。ACM 从 2008 年开始主办了专门的推荐系统国际会议（ACM Conference on Recommender System）。ACM SIGKDD 国际会议连续两年在其年度赛事 KDDCUP 中使用推荐系统作为竞赛主题（KDD’11 和 KDD’12）。与

此同时，产业界对推荐系统的研究也非常积极。很多公司公开了他们的数据集供大家研究，其中比较著名的有 Netflix 举办的推荐系统大赛，百度也在 2013 年举办了“百度电影推荐系统算法创新大赛”。此外，一些实际的推荐系统已经在为人们的日常生活提供服务，例如著名电商网站亚马逊的推荐系统向用户推荐其可能感兴趣的产品，为亚马逊提供了至少 35%的销售业绩[2]；著名 DVD 租赁网站 Netflix 使用推荐系统辅助用户寻找可能感兴趣的电影，为 Netflix 提供了大约 60%的租赁记录[3]。这些推荐系统通过数据分析和挖掘的方式利用海量的用户行为数据，改善了用户的使用体验，有效提升了企业的销售业绩。如今，推荐系统的研究和应用已经涉及到人们生活中的各个领域，包括图书、新闻、电子商务等。表 1.1 列举了各领域中的著名推荐系统。

表 1.1 著名推荐系统举例

领域	推荐系统	网址
图书	亚马逊[6]	http://www.amazon.com
	当当网	http://www.dangdang.com
	豆瓣读书	http://book.douban.com
新闻	GoogleNews[7, 8]	http://news.google.com
	Genieo	http://www.genieo.com
	Digg	http://digg.com
电影	Netflix[9]	http://www.netflix.com
	Jinni	http://www.jinni.com
	MovieLens	http:// movielens.org
电子商务	亚马逊	http://www.amazon.com
	淘宝	http://www.taobao.com
	京东	http://www.360buy.com
音乐	豆瓣电台	http://www.douban.fm
	Ringo[10]	http://ringomo.com
	Lastfm[11]	http://www.lastfm.com
	Pandora	http://www.pandora.com
视频	Hulu[12]	http://www.hulu.com

[2] http://glinden.blogspot.com/2006/12/35-of-sales-from-recommendations.html
[3] http://www.wired.com/wired/archive/12.10/tail.html

	Youtube[13, 14]	http://www.youtube.com
	Clicker	http://www.clicker.com
文章	StumbleUpon	http://www.stumbleupon.com
	CiteULike	http://www.citeulike.com
旅游[15-18]	TripAdvisor	http://www.tripadvisor.com
	NileGuide	http://www.nileguide.com
	YourTour	http://www.yourtour.com
社交网络	Facebook	http://www.facebook.com
	Twitter	http://twitter.com
	新浪微博	http://weibo.com
	人人网	http://www.renren.com

一个完整的推荐系统由三个部分组成：收集用户信息的行为采集模块，分析用户兴趣的模型分析模块以及推荐算法模块[19]。其中，行为采集模块负责记录用户的行为信息，包括浏览、下载、评分、购买、收藏等；模型分析模块利用采集到的用户信息，分析用户对不同产品[4]的喜好程度，并结合适合当前应用的模型分析用户兴趣；最后由推荐算法结合用户兴趣和产品特征向用户进行产品推荐。推荐算法是其中最为核心的部分。

经过多年的研究，研究人员提出了各种类型的推荐算法。内容过滤（Content-based Filtering）推荐算法[19]认为用户会喜欢和他以前喜欢的产品内容上相似的产品。该算法主要通过分析和利用产品的内容信息寻找相似的产品，比如一部电影的导演、演员、电影类型等，一本书的作者、标题、学科、关键字、出版商等。协同过滤（Collaborative Filtering）推荐算法[5, 6, 20, 21]不需要对产品的内容进行分析，而是利用大量用户的行为数据，从中找到特定的行为模式，以此预测用户的未来行为，并做出推荐。近年来，随着社交网络的兴起，社会化过滤（Social Filtering）推荐算法[2, 22]成为推荐系统研究的一个新领域。这类推荐算法主要利用社交网络中的好友关系，向用户推荐好友，或者推荐其好友感兴趣的产品。除此之外，基于人口统计学的推荐算法[23, 24]、基于位置的推荐算法[25, 26]等也在推荐系统中得到了广泛的研究和应用。

其中，协同过滤推荐算法使用群体智能发掘用户潜在兴趣，无需对产品内容进行

[4] 产品是指推荐系统中的待推荐对象，对应英文中的 Item。常见的中文表述方式有产品、物品和项三种类型，本论文使用产品进行表述。

语义分析，具有良好的领域适用性和推荐新颖性[20, 21, 27]，受到了研究人员的重点关注，相关技术已获得了巨大的成功和较为广泛的应用。但是，随着互联网的飞速发展，用户和产品数量暴增，协同过滤推荐技术面临的环境时刻都在发生变化，存在一些亟待解决的问题，主要包括数据稀疏性、冷启动、流行偏置、可扩展性、动态性、准确率和多样性等。本论文正是针对其中的数据稀疏性和流行偏置问题进行研究，通过解决这些问题，提升协同过滤推荐算法的推荐准确率和多样性。

推荐系统的目标是帮助用户选择一些用户自己可能感兴趣的产品，其主要实现形式是在合适的场景为用户提供一个包含 N 个产品的推荐列表。因此，这 N 个产品是否符合用户的兴趣是推荐系统关心的核心问题之一，这就是 Top-N 推荐问题，也是本论文关注的目标问题。

综上所述，推荐系统是解决大数据时代信息过载问题的关键技术之一，具有重要的理论研究价值和实际应用价值。因此，该领域的研究受到了学术界和产业界的广泛关注。Top-N 推荐是推荐系统研究的核心问题之一，协同过滤推荐技术是推荐系统的重要解决方案。本论文主要针对协同过滤推荐技术面临的部分挑战进行研究，希望可以提升相关算法的 Top-N 推荐效果。

1.2 国内外研究现状及发展趋势

自 20 世纪 90 年代以来，推荐系统的研究已经有了 20 多年的历史。由于推荐系统具有重要的理论价值和实用价值，该领域的研究受到了学术界和产业界的广泛关注。在学术界， ACM 从 2008 年开始主办专门的推荐系统国际会议（ACM Conference on Recommender System）；计算机领域的诸多国际顶级会议（包括 SIGKDD、SIGIR、SIGCHI、AAAI、ICDM 等）每年都有推荐系统方面的研究成果发表；SIGKDD 会议曾连续两年在其年度赛事 KDDCUP 中使用推荐系统作为竞赛主题（KDD’11 和 KDD’12）。在产业界，已有诸多实用推荐系统服务于互联网新闻、电子商务、社交网络等各个领域；各大互联网公司（包括谷歌、亚马逊、百度、阿里、腾讯等）也都有专门研究推荐系统的团队。

推荐算法是推荐系统研究的核心技术，协同过滤推荐技术是一类重要的推荐算法。本论文主要针对推荐系统中的 Top-N 推荐问题，进行协同过滤推荐技术的研究。本节首先从总体上对推荐算法在国内外的研究现状及发展趋势进行介绍；其次，按照学习目标的不同，详细介绍两种类型的协同过滤推荐技术，分别是以预测用户对产品

评分值为目标的评分预测推荐算法和以预测用户对产品喜好程度排序结果为目标的排序预测推荐算法；接下来，从准确率评价和多样性评价两个方面分别介绍主流的推荐算法评价指标；最后，对协同过滤推荐中面临的主要挑战进行分析和总结，并引出本论文的研究目标。

1.2.1 推荐算法

推荐算法一直以来是推荐系统领域研究的核心问题。按照使用数据的不同，推荐算法可以分为协同过滤、内容过滤和社会化过滤等类型[2]，各类推荐算法具体介绍如下。

(1) 协同过滤推荐算法

协同过滤（Collaborative Filtering）推荐算法是指主要通过利用用户的历史行为信息为用户建模进而做出推荐的一类算法。最初的协同过滤推荐算法是基于用户的协同过滤推荐算法（User-based Collaborative Filtering）[5]。该算法认为，一个用户会喜欢和他有相似爱好的用户喜欢的产品。因此，他们利用相似用户的信息进行近邻搜索，然后基于近邻用户的行为信息预测用户兴趣，向用户推荐产品。基于产品的协同过滤算法（Item-based Collaborative Filtering）是另一种经典的协同过滤算法，它认为一个用户会喜欢和他之前喜欢的产品类似的产品[6]。该算法主要利用相似产品的信息进行近邻搜索和用户行为预测，然后依据预测结果向用户推荐产品。这两种算法都是利用近邻信息进行推荐的协同过滤算法，因此被称为基于近邻的协同过滤推荐算法（Neighbor-based Collaborative Filtering）。

基于近邻的协同过滤推荐算法是一种启发式的推荐算法，无需预先训练推荐模型，也无法获取数据中的潜在知识。基于模型的推荐算法（Model-based Collaborative Filtering）是另一类典型的协同过滤推荐算法。它们的基本思想是使用用户历史信息训练一个推荐模型，然后利用这个模型对用户行为进行预测和推荐[20, 21]。典型的基于模型的推荐算法包括：贝叶斯网络模型[21]，聚类模型[28-30]，极大熵模型[31]，线性回归模型[32]，主题模型[33, 34]，矩阵分解模型[35-37]，图模型[38-41]等。

协同过滤推荐算法是本论文研究的核心问题。依据学习目标的不同，协同过滤推荐算法可以分为评分预测推荐算法和排序预测推荐算法两大类。本论文将在 1.2.2 节和 1.2.3 节分别对其进行详细介绍。

(2) 内容过滤推荐算法

内容过滤推荐算法是推荐系统研究的一个重要分支，最初的内容过滤推荐算法（Content-based Filtering）是协同过滤技术的延续与发展，它不需要用户对产品的评价意见，而是依据产品的内容信息计算用户或产品之间的相似性，进而完成相应的推荐任务[19] [42]。因此，其推荐效果主要取决于对产品内容特征表达和抽象的效果。

内容过滤推荐算法的核心在于对用户和产品建立配置文件，建立产品配置文件又是其中的核心问题。内容过滤推荐算法的基本流程为：首先利用机器学习等技术分析产品的内容，建立产品配置文件；然后根据用户访问历史信息及相关产品的配置文件聚合建立用户配置文件；最后按照用户配置文件和产品配置文件的相关程度进行推荐[43]。向量空间模型[44]是产品建模中最著名的模型。该模型首先抽取产品关键词，然后利用 TF-IDF（全称为 Term Frequency–Inverse Document Frequency）来计算关键词权重，使用加权的关键词向量来表示产品模型[45]。

从文本信息中获取关键词已经是比较成熟的技术，但是面对互联网中广泛存在的多媒体信息，如何准确的抽取关键词是个困难的问题。针对这个问题，有研究者使用人工标注的标签辅助进行关键词的抽取[46-48]。除此之外，内容过滤推荐算法在扩展用户兴趣方面还存在较大的局限性，即内容过滤推荐算法只能推荐与用户历史兴趣相关的产品，很难发现用户的潜在兴趣，实现兴趣的跳跃式推荐。

(3) 社会化过滤推荐算法

在现实社会中，用户的兴趣和选择往往受到好友的影响。近年来，随着诸如 Facebook，Twitter 等社交网络网站的兴起，如何利用用户之间的社会关系设计推荐系统，成为了推荐系统领域新兴的研究问题[2, 22, 49]。

社会化过滤（Social Filtering）推荐算法利用用户的社会关系向用户进行推荐，最简单的社会化过滤算法是基于近邻的算法。不同于传统的协同过滤算法利用用户评分信息寻找最近邻，社会化过滤推荐算法主要利用用户在社交网络中的关系寻找最近邻，然后利用最近邻的历史行为进行信息推荐[50-52]。研究表明，经过朋友推荐的东西被接受和好评的概率都远远大于平均情况[53, 54]。此外，利用用户的社会关系，可以一定程度上解决协同过滤推荐的冷启动和稀疏性问题。比如对于新用户，可以通过 Facebook 等网站提供的 API 获取他的社交网络信息，然后根据他的朋友的历史行为，为其预测兴趣和推荐产品。但是，社会化过滤也有一些缺点。用户之间的社会关系形成原因很多，可能是因为血缘关系、同学关系或者有共同的兴趣爱好。但只有依赖共

同兴趣爱好形成的社交网络关系才会对预测用户的兴趣有较大的帮助。因此，如何鉴别不同社会关系对预测用户不同行为的作用，是社会化过滤中的一个重要研究方向[2]。另外，社会化过滤可能会带来用户隐私性的问题，如何在保护用户隐私性的前提下有效地进行推荐也是社会化过滤的一个研究难点[55]。

除了利用社交网络信息给用户推荐产品的社会化过滤问题，向用户推荐好友也是社会化过滤的重要研究方向[56-59]。主要的社交网络系统都有好友推荐的功能，如Facebook，新浪微博等。

(4) 其它推荐算法

除了以上三类推荐算法，实际系统中还存在其它类型的推荐算法。关联规则作为数据挖掘领域的经典算法之一[60]，非常适合于推荐系统的应用场景。基于关联规则的推荐系统根据用户的历史信息进行规则提取，寻找有意义的关联组合，通过这些组合关系为用户生成推荐结果。传统的关联规则挖掘算法如 Apriori[61, 62]，FP 增长[63]等都可以直接应用到推荐系统领域。基于关联规则的推荐算法具有良好的可扩展性和可解释性[64]，并且可提供和基于近邻的协同过滤推荐算法类似的推荐准确率[65]，但是它的推荐结果的多样性往往比较差。一些研究者通过使用局部支持度代替全局支持度来改善这一问题，并取得了良好的效果[65] [66-68]。

基于人口统计学（Demographic）的推荐也是一种应用较为广泛的推荐算法。该算法利用用户的人口统计学特征，包括年龄、性别、工作、学历、国籍、民族、居住地等信息，将用户分类或聚类，然后利用用户所处的类别中其他用户的爱好进行推荐[23, 24]。基于人口统计学的推荐主要存在两个方面的问题：一是推荐粒度太粗，具有相同人口统计学特点的人获得的推荐结果往往是一样的[2]；二是用户有时出于隐私性的考虑不会填写真实的人口统计学信息，这就会带来推荐效果的大幅下降[69]。

随着移动互联网的发展和智能移动终端的普及，推荐系统可以方便的获得地点等重要的用户行为的上下文信息。有效利用用户行为的上下文信息可以提高推荐算法的推荐效果[70]。因此，最近几年位置过滤（Location-based Filtering）推荐算法也逐渐成为推荐系统领域研究的一个重要方向[25, 26, 71]。

以上介绍的几种类型的推荐算法都是单一的推荐算法。单一的推荐算法都有着不同的优缺点，且具有很强的互补性，因此，使用混合的方法结合不同算法进行推荐是一种有效的方式[72, 73]。混合推荐算法可以是不同类型推荐算法的混合，也可以是同种

类型不同推荐算法的混合。混合推荐算法主要包括两种形式：混合算法和集成学习。

混合算法往往是两种或多种算法的结合，可以达到取长补短、优势互补的效果。混合算法主要包括 7 种不同的混合形式[72, 74]：

- 加权（Weight）：为多种推荐算法设置不同的权重，然后加权累加各种算法的结果，并依此生成最终的推荐结果[75, 76]；
- 变换（Switch）：推荐系统根据问题背景和实际情况变换使用不同的推荐策略，但每次只根据具体的环境采取其中的一种策略[77, 78]；
- 混合（Mixed）：同时使用多种推荐技术得到多种推荐结果进行组合呈献给用户，为用户提供参考。该方法同加权的方法的不同之处在于无需对各个推荐结果进行加权，而是将各算法的推荐结果组合为一个整体的推荐结果列表，并将其返回给用户[79, 80]；
- 特征组合（Feature Combination）：组合来自不同推荐算法的数据源特征，将其应用到一种推荐算法中[81]；
- 层叠（Cascade）：一般使用两种或者多种推荐技术，首先使用一种推荐技术得到一个相对粗糙的结果，然后使用其它推荐技术对前一次的推荐结果进行过滤，得到更精确的结果[82]；
- 特征扩充（Feature Augmentation）：将一种推荐技术产生附加的特征信息嵌入到另一种推荐技术的特征输入中，以特征扩充的形式提升后者的推荐效果[83, 84]；
- 元级别（Meta-level）：该方法类似于特征扩充法，区别在于元级别直接使用一种推荐方法产生的模型而非特征信息作为另一种推荐方法的输入[85, 86]。

集成学习（Ensemble Learning）[87]通过训练多个学习器解决原始问题，可以有效提升系统的泛化学习能力，是机器学习（Machine Learning）的重要研究方向。在推荐系统研究领域，集成学习推荐算法通过将多种给定参数的单一推荐算法放在一起，再进行一轮学习，获取一种混合的推荐算法[88-90]。经过集成学习的算法预测准确率可以远超任何单个算法，2009 年 Netfilx 竞赛中获得冠军的算法即是综合了 453 种单独算法的集成学习推荐算法[89]。

1.2.2 评分预测推荐算法

协同过滤推荐算法是推荐系统领域的一类重要算法，已经被学术界和产业界进行了深入的研究，有着重要的学术研究意义和实用价值。评分预测推荐算法是协同过滤

算法的一个重要研究分支，自 20 世纪 90 年代以来一直是领域内关注的焦点。

在一个典型的推荐系统应用场景中，用户可以选择产品并进行评分。推荐系统会依据用户对已评分产品的评分情况，分析用户兴趣，并向用户推荐他可能感兴趣的其它产品。因此，以预测用户对未评分产品的评分值来预测用户对这些产品的兴趣程度就成为了一种经典的推荐解决方案。这就是评分预测推荐算法。

评分预测推荐算法利用已有用户评分数据，预测用户对未评分产品的评分值，并进行产品的推荐。假设 $R=\{<u,i,r>\}$是包含所有已知用户评分情况的集合，其中的每个元素$<u,i,r>$代表用户 u 对产品 i 的评分值为 r；U 是所有用户的集合，$U=\{u_1,u_2,\cdots,u_m\}$；I 是所有产品的集合，$I=\{i_1,i_2,\cdots,i_n\}$。对于任意用户 u，都存在一个向量 $I(u)$，是包含该用户已评分产品的集合；对于任意产品 i，存在一个集合 $U(i)$，包含所有已经对该产品评分的用户。评分预测推荐算法的任务包含两个部分[32]：

- 首先是利用 R 中的数据信息预测用户对未评分产品的评分。即对于用户 u，评分预测推荐算法需预测其对任意产品 $i \in I - I(u)$ 的评分 $\hat{r}(u,i)$。
- 然后依据未评分产品的评分预测值 $\hat{r}(u,i)$ 进行排序，将其中评分预测值较高的产品推荐给用户。

按照推荐方法的不同，评分预测推荐算法可以分为基于存储的协同过滤推荐算法（Memory-based Collaborative Filtering）和基于模型的协同过滤推荐算法（Model-based Collaborative Filtering）两个大类[20, 27]，下面将分别对它们进行具体介绍。

(1) 基于存储的协同过滤推荐算法

基于存储的协同过滤推荐算法一般都需要将用户的历史信息存入内存，然后在推荐时实时分析内存中的用户历史信息完成推荐任务。GroupLens 小组在 1994 年的 CSCW 国际会议中提出的基于用户的协同过滤推荐算法（User-based Collaborative Filtering，简称 UserCF）[5]是一种经典的基于存储的协同过滤推荐算法。该算法认为一个用户会喜欢和他有相似兴趣爱好的用户喜欢的产品。因此，要对一个用户做推荐，首先得找到和他兴趣爱好相似的用户。在 UserCF 中，两个用户兴趣爱好相似是因为他们喜欢相似的产品。这种相似性通过用户相似度进行衡量。衡量两个用户的相似度主要有两种思路：一种认为对于给定用户 u、a，若他们对于任意产品 i 总是给出相似的评分，则认为这两个用户相似，这种方法被称为 Correlation 相似度方法；另一种则认为如果用户 u、a 总是对相同的产品进行浏览、评价等行为，则这两个用户相似，

这种方法被称为 Relevance 相似度方法[21, 91]。下面我们将列举几种典型的衡量用户相似度的方法：

1) 余弦相似度（Cosine）是一种典型的 Correlation 相似度方法。它将用户的历史评分信息看作是 n 维向量，即使用 $\vec{u}$ 、$\vec{a}$ 分别表示用户 u 和用户 a 的历史评分信息。其中向量的第 i 个元素是该用户对第 i 个产品的评分值，未评分产品用 0 代替。用户 u 和用户 a 的余弦相似度可以用两个向量的夹角余弦表示，即：

$$sim_{\cos}(u,a) = \cos(\vec{u},\vec{a}) = \frac{\vec{u}\cdot\vec{a}}{\|\vec{u}\|\cdot\|\vec{a}\|} = \frac{\sum_{i\in I(u,a)} r_{ui}\cdot r_{ai}}{\sqrt{\sum_{i\in I(u)} r_{ui}^{\,2}}\cdot\sqrt{\sum_{i\in I(a)} r_{ai}^{\,2}}} \tag{1.1}$$

其中，r_{ui} 是用户 u 对产品 i 的评分值，$I(u,a)$ 是用户 u 和用户 a 共同评分的产品集合。

2) 皮尔逊相似度（Pearson Correlation）亦是一种典型的 Correlation 相似度方法。它是自然科学领域中广泛用于度量两个变量间线性相关程度的方法之一。在 UserCF 中，它可以有效描述两个用户在若干个产品上评分变化趋势的一致程度。其计算方法如公式(1.2)所示：

$$sim_{Pearson}(u,a) = \frac{\sum_{i\in I(u,a)} (r_{ui} - \overline{r(u)})\cdot(r_{ai} - \overline{r(a)})}{\sqrt{\sum_{i\in I(u,a)} (r_{ui} - \overline{r(u)})^2}\cdot\sqrt{\sum_{i\in I(u,a)} (r_{ai} - \overline{r(a)})^2}} \tag{1.2}$$

其中，$\overline{r(u)}$ 是用户 u 对产品的平均评分值。

3) Jaccard 相似度是一种典型的 Relevance 相似度方法。它通过计算用户 u 和用户 a 评分的产品集合的相似程度衡量两个用户之间的相似度，两个用户共同评分的产品越多则他们越相似，其计算方法为：

$$sim_{Jaccard}(u,a) = \frac{|I(u)\cap I(a)|}{|I(u)\cup I(a)|} \tag{1.3}$$

4) 对数似然相似度（Log-Likelihood）亦是一种典型的 Relevance 相似度方法。它通过计算用户 u 和用户 a 所评分产品集合的对数似然相似度衡量两个用户间的相似程度，其计算方法如公式(1.4)至(1.6)所示：

$$sim_{Log-Likelihood}(u,a) = 2\sum_{i,j} K_{ij}\cdot\log\frac{n_{ij}}{m_{ij}} \tag{1.4}$$

$$n_{ij} = \frac{K_{ij}}{K_{i1} + K_{i2}} \tag{1.5}$$

$$m_{ij} = \frac{K_{1j} + K_{2j}}{\sum_{i,j} K_{ij}} \tag{1.6}$$

其中，K_{ij}的取值如表 1.2 所示：

表 1.2 对数似然相似度参数定义

	User *u* rates	User *u* does not rate
User *a* rates	K_{11}	K_{12}
User *a* does not rate	K_{21}	K_{22}

利用计算所得的用户相似度，UserCF 为待推荐用户寻找近邻，以便利用近邻行为预测当前用户的行为。近邻搜索是 UserCF 算法的核心内容之一，其效率和质量直接影响推荐算法的有效性。近邻搜索往往需要为当前用户寻找 *K* 个最相似的用户，因此，亦被称为 *K* 近邻方法（K-Nearest Neighbors，简称 KNN）。

在确定了用户 *u* 的近邻集合后，UserCF 利用这些近邻的评分信息，将其进行加权平均，预测用户 *u* 对未评分产品的评分值。其计算方法如公式(1.7)所示：

$$\hat{r}_{UserCF}(u,i) = \overline{r(u)} + \frac{\sum_{a \in N(u)} sim(u,a) \cdot (r_{ai} - \overline{r(a)})}{\sum_{a \in N(u)} | sim(u,a) |} \tag{1.7}$$

其中，*sim(u,a)*为用户 *u* 和用户 *a* 的相似度，*N(u)*为用户 *u* 的近邻集合。在 Top-*N* 推荐中，UserCF 通过预测用户对产品的评分值信息，对用户未评分产品进行排序，并将预测评分值较高的前 *N* 个产品推荐给用户。

和基于用户的协同过滤推荐算法不同，基于产品的协同过滤推荐算法（Item-based Collaborative Filtering，简称 ItemCF）认为用户会喜欢和他之前喜欢的产品相似的产品[6]。因此，ItemCF 首先从用户历史行为信息中获取用户喜欢的产品，然后利用产品相似度信息寻找和这些产品相似的产品，并将其推荐给用户。ItemCF 的核心是计算产品的相似度。与 UserCF 的思想类似，ItemCF 认为两个产品被越多的人同时喜欢，两个产品就越相似。ItemCF 中产品相似度计算方法与 UserCF 中的用户相似度计算方法类似，只是计算相似度的目标对象变为产品，并相应的利用不同用户对同一产品的评分向量作为数据来源，其计算细节在此不再赘述。ItemCF 利用产品相似度为目标产品寻找最近邻，依据最近邻的评分信息预测用户对目标产品的评分，并依此进行推荐。ItemCF 的评分预测方法如公式(1.8)所示：

$$\hat{r}_{ItemCF}(u,i) = \overline{r(i)} + \frac{\sum_{j \in I(u) \cap N(i)} sim(i,j) \cdot (r_{uj} - \overline{r(j)})}{\sum_{j \in I(u) \cap N(i)} | sim(i,j) |} \tag{1.8}$$

其中，$\overline{r(i)}$是所有用户对产品 i 的平均评分，$N(i)$是产品 i 的近邻产品集合，$sim(i,j)$是产品 i 和产品 j 的相似度。

UserCF 和 ItemCF 是两种典型的基于存储的协同过滤推荐算法。对比两者，UserCF 主要利用和用户兴趣相似的用户群体的行为信息进行推荐，它的推荐结果较为社会化；而 ItemCF 是向用户推荐与其自身历史兴趣相关的产品，它的推荐结果更加个性化[4]。因此，这两种方法分别适用于需要较社会化或者较个性化推荐结果的应用环境。此外，UserCF 和 ItemCF 分别需维护一个用户或产品的相似度矩阵，从技术实现的角度，两者分别适应于用户或产品相似度较为稳定的环境。在电子商务、在线视频网站等推荐系统应用广泛的平台中，产品相似度相对稳定，因此，ItemCF 应用也较为广泛。ItemCF 还有一个特点是可以进行推荐解释。如图 1.4 所示，Amazon 系统向某用户推荐一本《驾驭大数据》的书，并向用户解释系统推荐这本书是因为该用户曾购买了《大数据•互联网大规模数据挖掘与分布式处理》、《推荐系统实践》和《社交网站的数据挖掘和分析》三本书。从解释的方式可以看出，Amazon 系统使用了 ItemCF 的推荐方法[5]。这种解释为用户带来了更好的体验性。有研究表明，好的推荐解释可以有效提高用户对推荐结果的接受程度[92, 93]。

UserCF 和 ItemCF 分别利用用户或产品的近邻信息，向用户推荐相关产品，因此，又被称为基于近邻的协同过滤推荐算法（Neighbor-based Collaborative Filtering）。除此之外，Slope-one 也是一种基于存储的协同过滤推荐算法。该算法是一种非常简洁的基于产品评分差异的推荐算法[94]。它的基本假设是对于任意产品 i，j，若大家对 i 和 j 的平均评分分差为 x，则当前用户对这两个的产品评分分差也很可能是 x。Slope-one 根据该假设对用户的未评分产品进行评分预测和推荐[94-96]。

[5] Amazon 利用该用户的历史购买行为，向其推荐与其中某些产品类似的产品，是典型的 ItemCF 推荐方法。

图 1.4 Amazon 系统 ItemCF 推荐解释

(2) 基于模型的协同过滤推荐算法

基于存储的协同过滤推荐算法将用户的历史行为信息放入内存，利用启发式的方法寻找用户可能感兴趣的产品，并将其推荐给用户。这类方法在各推荐系统中得到了广泛的应用，并获得了良好的效果。但是，随着互联网数据的爆发式增长，用户、产品的规模都迅速扩大，用户评分矩阵变得更为稀疏，使得传统的基于存储的协同过滤推荐算法遇到了有效性和实时性等各方面的挑战。为解决这个问题，有研究者提出了一系列基于模型的协同过滤推荐算法，这些推荐算法利用用户历史评分数据训练推荐模型，然后利用训练好的推荐模型预测用户评分，并依此进行推荐。典型的推荐模型包括矩阵分解模型（Matrix Factorization）[35-37, 97]、主题模型（Topic Model）[98-101]等。

基于矩阵分解模型的推荐算法的主要思想是通过降维技术，将用户-产品评分矩阵近似表示到低维空间，然后利用降维后的矩阵对用户评分进行预测和推荐。一个基本的矩阵分解模型如公式(1.9)所示：

$$R \approx PQ^T \tag{1.9}$$

对于$m \times n$维的用户-产品评分矩阵R，矩阵分解模型利用矩阵分解技术将其近似表示为一个$m \times k$维的用户矩阵P和一个$n \times k$维的产品矩阵Q的乘积形式。其中，m是用户个数，n是产品个数，k是矩阵分解模型描述用户和产品潜在因子个数的参数。

由于推荐系统中的用户评分矩阵R是稀疏矩阵，因此，我们不能直接使用奇异值分解（Singular Value Decomposition）等数学方法进行矩阵分解的计算。矩阵分解模型推荐算法往往使用梯度下降法，以最小化分解后的近似矩阵与原始用户评分矩阵的误差为目标，进行训练[102, 103]，其目标函数为：

$$\min_{p*,q*} \sum_{<u,i>} (r_{ui} - p_u \cdot q_i^T)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2) \tag{1.10}$$

其中，r_{ui}为原始评分矩阵中的用户u对产品i的评分值，p_u、q_i分别为用户u和产品i在低维矩阵P、Q中的对应向量，λ是避免过拟合的正则化参数。训练好的矩阵分解模型利用低维矩阵P和Q预测用户对未评分产品的评分值，预测方法如公式(1.11)所示：

$$\hat{r}(u,i) = p_u \cdot q_i^T \tag{1.11}$$

典型的基于矩阵分解的推荐算法包括奇异值分解（Singular Value Decomposition）[37, 104-106]，最大边际矩阵分解（Maximum Margin Matrix Factorization）[107]，非负矩阵分解（Non-negative Matrix Factorization）[108]等。这种类型的推荐算法通过降维技术降低了实时推荐的计算复杂性，有较好的扩展性，同时还可以得到用户和产品间的潜在联系。

主题模型是文本处理领域应用非常广泛的模型之一。它是一种对文档隐含主题进行建模的方法，可以有效描述一篇文档（Document）包含多个主题（Topic）的特征。在推荐系统领域，一个用户往往有着多个不同的兴趣方向，一个产品也可能属于不同的类别。这样的特点非常适合使用主题模型进行建模。实际上，主题模型也是近年来推荐系统研究领域关注的重点之一，已经有诸多研究成果发表[98, 99, 109, 110]。

前文所述的矩阵分解模型就可以视为一种主题模型。矩阵分解模型亦称为潜语义模型（Latent Semantic Analysis，简称 LSA），它通过矩阵分解的技术将用户评分行为信息投影到潜语义空间，分解后的用户向量p_u可以理解为用户u在各潜语义上的偏好，而产品向量q_i可以理解为产品i对各潜语义的隶属程度。

概率潜语义模型（Probabilistic Latent Semantic Analysis[99]，简称 PLSA，亦称为 Probabilistic Latent Semantic Indexing[109]，简称 PLSI）是矩阵分解模型的扩展。它使用概率模型描述文档包含多个主题的特征。Hoffman 将该模型应用到推荐系统领域[100]，将用户描述为多个不同兴趣主题的联合分布，每个兴趣主题中各产品有着不同的概率分布。用户对产品的评分是各兴趣主题上用户对产品评分的联合概率分布。该模型的形式化描述如公式(1.12)所示：

$$P(r \mid u, i) = \sum_{z} P(r \mid i, z) \cdot P(z \mid u) \tag{1.12}$$

具体而言，PLSA 算法认为用户 u 对产品 i 评分为 r 的概率 $P(r \mid u, i)$ 依赖以下计算过程：

1) 计算用户以一定概率选择主题 z（即 $P(z \mid u)$），并且在主题 z 上对产品 i 评分为 r（即 $P(r \mid i, z)$）的概率；
2) 对所有主题中用户 u 对产品 i 评分为 r 的概率进行累计。

PLSA 模型通常使用 EM（Expectation Maximization）算法，以最大化观测数据的产生概率为目标进行训练，进而获取各条件概率分布的取值。其目标函数如公式(1.13)所示：

$$\max \prod_{<u,i,r>} P(r \mid u, i) \tag{1.13}$$

利用训练好的模型参数，PLSA 使用公式(1.14)预测用户对产品的评分期望，并将评分期望较大的产品推荐给用户。

$$E(u, i) = \sum r \cdot P(r \mid u, i) \tag{1.14}$$

文献[98]在 PLSA 的基础上，认为用户对不同主题感兴趣的概率服从狄利克雷分布（Dirichlet Distribution），提出了 LDA（Latent Dirichlet Allocation）模型，并将其应用到了推荐系统领域。LSA、PLSA 和 LDA 都是推荐系统领域中使用较为广泛的主题模型。

基于模型的协同过滤推荐算法除了包含以上两大类算法外，还包括贝叶斯网络模型[21]，聚类模型[28-30]，极大熵模型[31]，线性回归模型[32]，图模型[38, 39, 111-114]等。基于模型的协同过滤推荐算法往往建立在大量用户历史数据的基础上，通过统计学、机器学习等相关技术，学习一个有效的推荐模型，利用该模型向用户推荐产品[20]。这类算法相较于基于存储的协同过滤推荐算法有着更好的理论基础[4]。

评分预测推荐算法在推荐系统的发展过程中贡献巨大，已经广泛应用在不同领域

的推荐系统中。但是，推荐系统的目标并不是预测用户对产品的评分值，而是为用户选择他们可能感兴趣的产品，并以便捷的形式呈现给用户。这就是 Top-*N* 推荐问题，也是本文研究的目标问题。有研究表明，评分预测的准确程度和 Top-*N* 推荐效果之间并没有直接关系[115, 116]。例如，对于用户真实评分分别为 3 分和 4 分的两个产品，推荐系统 A 预测用户评分分别为 2 分和 5 分，而推荐系统 B 预测的评分值分别为 4 分和 3 分。则两个推荐系统有着相同的评分预测准确率，但是，两者向用户进行 Top1 推荐的效果则截然相反[116, 117]。也就是说，简单的以提高评分预测准确率为优化目标，并不一定可以提升推荐系统的 Top-*N* 推荐效果。为解决这个问题，有研究者提出通过预测用户对产品的排序情况进行推荐的排序预测推荐算法。接下来将对排序预测推荐算法的研究情况进行详细介绍。

1.2.3 排序预测推荐算法

与评分预测推荐算法以预测用户对产品的评分为目标不同，排序预测推荐算法以预测用户对产品喜好程度的排序结果为目标，并依据预测的排序情况向用户推荐产品。这类推荐算法通常将用户-产品评分矩阵转变为用户对产品喜好程度的排序信息，以最佳拟合该排序信息为训练目标，完成推荐模型的训练。排序预测推荐算法是近年来兴起的研究方向，相关研究相对较少，下面将列举其中主要的研究成果。

文献[118]认为评分预测推荐算法是一种错误的推荐系统解决方案，好的推荐算法应直接以排序效果为优化目标。此外，他们还认为排在前面的产品的排序结果正确率的重要程度要远胜于排在后面的产品。因此，他们选择和排序位置相关的 NDCG 效果（全称为 Normalized Discounted Cumulative Gain）作为优化目标，使用最大边际矩阵分解模型（Maximum Margin Matrix Factorization）训练推荐模型。该方法被称作 CofiRank 推荐算法，其优化目标是一种列表序（Listwise）的排序目标函数，具有良好的可扩展性。

文献[116]同样认为排序预测是一种相较于评分预测更有效的推荐算法。他们使用用户对产品对的排序顺序作为计算用户相似度，以及为用户寻找近邻的依据。基于选择的用户近邻，他们利用贪心策略（Greedy Strategy）或者随机游走模型（Random Walk Model）使用近邻信息，进行排序预测，并依此向用户推荐产品。该方法是一种对序（Pairwise）的排序预测推荐算法。同样是利用对序的排序目标，文献[117]提出了一种基于概率主题模型的排序预测推荐算法，称为 PLPA 算法（全称为 Probabilistic Latent Preference Analysis）。PLPA 算法利用 Bradley-Terry 模型描述用户对产品对的偏好情况，

使用概率主题模型描述用户在不同主题上对产品的偏好情况。该算法是 PLSA 推荐算法的变种。PLSA 模型描述的是用户在不同主题上对特定产品的评分，而 PLPA 模型描述的是用户在不同主题上对特定产品对的偏好情况。

使用对序目标的排序预测推荐算法应用较为广泛。文献[119-121]中分别使用不同的对序目标函数解决单类推荐[122]问题。其中，文献[119]使用 AUC（全称为 the Area Under the Curve）作为优化目标，文献[120]使用 NDCG 作为优化目标，文献[121]使用 MRR（全称为 Mean Reciprocal Rank）作为优化目标。文献[123]提出了在产品集合上设置对序目标进行优化的 CoFiSet 算法。该算法利用用户反馈信息，针对每个用户将产品分为有用户反馈的产品集合和无用户反馈的产品集合，认为有用户反馈的产品集合的平均预测值应大于无用户反馈的产品集合的平均预测值，并通过实验验证了在产品集合上设置对序目标的方法相对在产品对上设置对序目标的方法有着更好的稳定性。

文献[124]提出了一种基于点序（Pointwise）的排序预测推荐算法，并将其称作 OrdRec 算法。该算法将用户对产品的评分信息当作是用户对产品喜好程度进行排序的序数信息，即不同评分之间只有顺序关系，而无数字关系。OrdRec 算法首先预测用户对产品的评分，然后将该评分对应到序数的概率分布上，最后利用 Logistic 函数计算其排序值。该算法不仅可以预测用户对产品的具体评分情况，还可以预测他进行不同评分的概率分布情况，以及预测结果的置信度信息。

为了有效利用评分预测推荐算法和排序预测推荐算法的优点，文献[125]提出了一种对两者进行整合的统一推荐框架。在该框架中，评分预测推荐算法和排序预测推荐算法共享统一的用户和产品特征向量，通过线性组合的方式将评分预测准确率和排序预测准确率整合为一个统一的优化目标函数，使用梯度下降法训练推荐模型，并进行推荐。在实验中，他们使用 PMF 算法（全称为 Probabilistic Matrix Factorization）[126]进行评分预测，使用 ListRank 算法[127]进行排序预测。实验结果验证了整合后的推荐算法相比单一的评分预测推荐算法和排序预测推荐算法有着更好的 Top-*N* 推荐效果。该结论还说明了 Top-*N* 推荐亦不是简单的排序预测问题，评分预测在 Top-*N* 推荐中仍占有重要地位。

综上所述，虽然排序预测推荐算法比评分预测推荐算法更贴近 Top-*N* 推荐的本质，但是 Top-*N* 推荐并不仅仅是一个排序预测问题。因此，在推荐系统的研究中，需要继续对 Top-*N* 推荐的本质进行探索。本论文的第 3 章和第 4 章将分别探讨数据稀疏背景

下的缺失数据建模方法和用户行为模式，并针对性的提出可以有效提升推荐算法Top-*N*推荐表现的解决方案。

1.2.4 推荐效果评价

经过多年的发展，各种各样的推荐系统已经应用到各个领域中。如何判断一个推荐系统是不是一个好的推荐系统，得到了研究者们的广泛关注[128-133]。由于推荐系统的应用目标不同，往往需要不同的评价方法对推荐算法在当前目标下的表现进行评价，以帮助推荐系统的决策者选择适合当前环境和任务的推荐算法。对推荐效果的评价主要分为两种类型：准确率评价和多样性评价。

(1) 准确率评价

推荐结果的准确率一直以来是推荐算法关注的重点问题，推荐准确率的高低直接显示了推荐系统预测用户行为的能力。依据不同的预测目标，推荐结果的准确率评价方式主要分为评分预测准确率和Top-*N*推荐准确率两种类型。

评分预测准确率将系统预测的评分与用户的实际评分数据做对比，通过衡量它们的误差来评价推荐算法的准确率。针对这个问题，平均绝对误差（Mean Absolute Error，简称MAE）[32, 94, 134]和均方误差（Root Mean Squared Error，简称RMSE）[37, 105, 134]是最常用的两种评价方式。其计算方式分别如公式(1.15)和公式(1.16)所示：

$$MAE = \frac{\sum_{<u,i>\in T} \left| r_{ui} - \hat{r}(u,i) \right|}{\left| T \right|} \tag{1.15}$$

$$RMSE = \sqrt{\frac{\sum_{<u,i>\in T} (r_{ui} - \hat{r}(u,i))^2}{\left| T \right|}} \tag{1.16}$$

其中，T为测试集，$\hat{r}(u,i)$是推荐算法预测的用户u对产品的i的评分值，r_{ui}为测试集中用户u对产品i的实际评分。

MAE或者RMSE的取值越小说明推荐算法对用户的评分预测越准确。相对于MAE，RMSE以平方惩罚的形式放大了预测不准的评分结果的惩罚，对推荐系统的预测准确率要求更加苛刻。早期的研究主要使用MAE对推荐算法的评分预测准确率进行评价[94, 106]，但随着Netflix大赛将RMSE作为评价指标，RMSE逐渐成为评分预测问题中预测准确率的主要评价指标[37, 91, 105]。

Top-*N*推荐问题是指对每一个用户，通过分析他们的历史行为信息，为他们推荐

N 个他们可能喜欢的产品。相应的，Top-N 推荐准确率评价则需要衡量用户是否真正喜欢算法推荐的产品。准确率（Precision）和召回率（Recall）是针对 Top-N 推荐准确率的两种主要评价指标[115, 130, 135]，其计算方法如公式(1.17)和公式(1.18)所示：

$$Precision(N) = \frac{1}{|U|}\sum_{u\in U}\frac{L_N(u)}{N} \tag{1.17}$$

$$Recall(N) = \frac{1}{|U|}\sum_{u\in U}\frac{L_N(u)}{L(u)} \tag{1.18}$$

其中，U 是测试集中所有用户的集合，$L_N(u)$是针对用户 u 的 Top-N 推荐结果中用户 u 喜欢的产品，$L(u)$是测试集中用户 u 喜欢的所有产品。公式(1.17)和公式(1.18)中所述评价指标都与 Top-N 推荐的列表长度 N 相关。其中，准确率是 Top-N 推荐结果中用户喜欢的产品所占的比例，召回率是 Top-N 推荐结果中用户喜欢的产品占该用户所有喜欢产品的比例。

此外，K-Call[121, 136]亦是一种常用的 Top-N 推荐准确率评价指标。该指标通过统计 Top-N推荐结果中含有至少K个相关产品的用户所占比例评价推荐算法推荐相关产品的能力，其计算方法如公式(1.19)所示：

$$K\text{-}Call@N = \frac{1}{|U|}\sum_{u\in U} I(|L_N(u)| \geq K) \tag{1.19}$$

其中，函数 $I(\bullet)$是指示函数，即变量为真时函数取值为 1，为假时取值为 0。在实际应用 K-Call 时，往往选择 K=1 进行推荐效果评价，即使用 1-Call 评价推荐算法在 Top-N 推荐中包含至少一个相关产品的能力。

准确率、召回率和 K-Call 是与推荐顺序无关的 Top-N 推荐准确率评价方法。但是，有研究者认为用户喜欢的产品在推荐结果中出现的位置越靠前，算法的推荐准确率越高[137, 138]。他们按照用户喜欢的产品在推荐结果中出现的顺序对推荐算法进行准确率评价。NDCG 是一种应用广泛的、和顺序相关的 Top-N 推荐准确率评价指标，其计算方法如下：

$$DCG@N(u) = \sum_{p=1}^{N}\frac{2^{R(u,p)}-1}{\log(1+p)} \tag{1.20}$$

$$NDCG@N = \frac{1}{|U|}\sum_{u\in U}\frac{DCG@N(u)}{IDCG@N(u)} \tag{1.21}$$

其中，*R(u,p)*是用户 *u* 对推荐列表中第 *p* 个产品的评分，IDCG@N(u)是进行归一化的参数，其取值是 DCG@N(u)的最大值。NDCG 综合考虑了 Top-*N* 推荐结果与用户兴趣的相关程度以及推荐结果的排序情况，是一种全面的 Top-*N* 推荐准确率评价指标。但是，在推荐系统中，用户并不会对所有产品进行评分。尤其是进行离线评测时，Top-*N* 推荐结果中的很多产品并未被用户显性评分。这为我们计算 NDCG 指标带来了较大的困难。一种解决方案是将用户未评分产品当做无关产品，即令其对应的 *R(u,p)*取值为 0，进行 NDCG 的计算。本论文使用这种方案作为评价推荐算法 NDCG 表现的方法，该方法亦是本论文评价算法 Top-*N* 推荐准确率的主要指标。

有研究者将 Top-*N* 推荐问题转换为排序预测问题，并利用推荐算法对测试集中的产品排序的结果评价算法的推荐准确率[117]。他们使用 NDCG 评价推荐算法对测试集中产品排序的表现，本论文将这种使用 NDCG 评价指标的方案记为 NDCG+。NDCG+ 虽然不能直接反映推荐算法的 Top-*N* 推荐效果，但是本论文亦将其选作评价算法推荐效果的对照指标，使用它对推荐算法的排序预测能力进行评价。

评分预测问题一直是推荐系统研究的热点，大多数推荐系统领域的研究都是基于评分预测的。因此，评分预测准确率也受到了更多的关注。Netflix 大赛、百度推荐系统大赛等比赛也都是以评分预测准确率作为评价标准的。但是，Top-*N* 推荐才是更符合实际应用需求的评价指标[6]。近年来，学术界逐渐认识到了 Top-*N* 推荐的重要性。本论文顺应这个趋势，针对 Top-*N* 推荐问题进行重点研究。因此，我们在实验环节关注的重点是推荐算法的 Top-*N* 推荐准确率。

(2) 多样性评价

准确率是评价推荐算法优劣的重要指标。但如果推荐算法只能为用户推荐同一种类型的产品，即使准确率很高亦会影响用户的体验。因此，推荐丰富多彩的产品是推荐算法关注的另一个重要方面，推荐结果的多样性和新颖性也是评价算法推荐效果的重要指标。

一般而言，推荐系统多样性分为两个层次：用户内推荐结果多样性（Intra-user diversity）和用户间推荐结果多样性（Inter-user diversity）。用户内推荐结果多样性可以靠用户推荐列表中不同产品的相似性（IntraSim）[139]来衡量：

[6]亚马逊前科学家 Greg Linden 曾在博文“What is a Good Recommendation Algorithm?”中提出该观点。博文网址为 http://cacm.acm.org/blogs/blog-cacm/22925-what-is-a-good-recommendation-algorithm/fulltext

$$IntraSim = \frac{1}{|U|}\sum_{u\in U}\frac{\sum_{i,j\in Rec(u)} sim(i,j)}{N(N-1)} \tag{1.22}$$

其中，*Rec(u)*是用户 *u* 的 Top-*N* 推荐结果，*sim(i,j)*是产品 *i*、*j* 的相似度。IntraSim 越小，说明推荐结果的用户内推荐结果多样性越好。用户间推荐结果多样性（InterDiv）是通过对比不同用户推荐结果的区别来衡量的[139]：

$$InterDiv = \frac{1}{|U|(|U|-1)}\sum_{u,v\in U}(1-\frac{Q_{uv}}{N}) \tag{1.23}$$

其中，Q_{uv} 是用户 *u*、*v* 的 Top-*N* 推荐结果中共同出现的产品数。InterDiv 越大，说明不同用户推荐结果中共同出现的产品越少，相应的，用户间推荐结果的多样性越好。

此外，推荐结果对系统产品的覆盖程度在一定程度上也反映了推荐结果的多样性。它度量了一个推荐算法可以把多大比例的产品推荐给至少一个用户[131]。针对 Top-*N* 推荐问题，算法的覆盖度（COV）可以使用公式(1.24)进行计算：

$$COV(N) = \frac{N_d}{N} \tag{1.24}$$

其中，N_d 是在 Top-*N* 推荐中，推荐算法推荐给至少一个用户的产品的总数。COV 越大，则说明推荐算法的多样性越好。

如果推荐算法可以为用户推荐一些意外的而又符合用户兴趣的产品，将大大提高用户的体验性，这就需要推荐算法具有一定的新颖性[130]。但是，推荐结果是否具有新颖性是一种用户的主观认知，很难通过离线指标进行评价。大多数对推荐算法新颖性的分析都是使用基于用户调查的方法，通过用户的直接反馈来评价算法新颖性[133]。有研究者认为，推荐产品的热门程度可以一定程度上反应推荐结果的新颖性，越不热门的产品越能让用户觉得新颖[11]。因此，可以使用推荐结果对长尾产品[7]的覆盖量（CIL，全称为 Coverage in Long-tail）评价推荐算法的新颖性，其计算方法如公式(1.25)所示：

$$CIL(N) = \frac{NL_d}{N} \tag{1.25}$$

其中，NL_d 是长尾产品集合和 N_d 的交集。CIL 越大则说明推荐结果中的长尾产品越多，推荐结果的新颖性越好。

[7] 长尾产品是指推荐系统应用背景中大量非热门产品的集合。本论文在计算 CIL 时，长尾产品特指除了最热门的 20%的产品之外的所有产品。

准确率、多样性和新颖性都是评价推荐系统效果的重要指标。一个好的推荐系统既要能为用户找到他们所关心、感兴趣的产品，又可以帮助用户扩展兴趣，向其推荐多样化、有新意的产品。本论文在研究协同过滤推荐算法的 Top-*N* 推荐问题时，既关心算法的推荐准确率，同时亦关注推荐结果的多样性和新颖性。

1.2.5 协同过滤推荐技术面临的主要挑战

从国内外的研究现状可以看出，协同过滤推荐是一项复杂的技术。虽然经过多年的研究已经取得了不少进展，但总体上仍处于发展阶段。而且随着互联网及信息技术的飞速发展，信息过载现象愈加明显，协同过滤推荐技术中存在着诸多亟待解决的问题和挑战，主要包括：

1) 数据稀疏性。协同过滤技术的推荐效果主要取决于用户数据的多少。用户的评分数据越多，推荐效果越好。但是在实际系统中，海量的产品数量（如 Netflix 上有数万部电影，亚马逊上有数百万本书）使得每个用户只在少数产品上有评价信息，且随着用户规模和产品数量的不断扩大，用户-产品评分矩阵往往会变得愈加稀疏。此外，即使每个用户评分的产品数量都很高，但是由于产品的海量性，这些评分信息的分布会很不均匀（通常服从幂律分布[140]）。在超高维并且分布不均的稀疏矩阵下，描述用户兴趣、衡量用户相似性都变得极其困难，从而导致推荐结果质量低下。目前，解决数据稀疏性问题主要有两种思路，一种是通过降维技术进行特征提取，将原始数据映射到低维空间，使变换后的数据变得相对稠密[106, 141, 142]；另一种是使用分类、聚类技术将用户行为进行聚合，在聚合后的子空间上进行推荐[29, 143, 144]。

2) 冷启动。冷启动问题包含新用户问题（New User）和新产品问题（New Item）两种情况。新用户问题是指当一个用户新加入到推荐系统时，由于系统中没有该用户的历史信息，无法为其提供推荐服务。新产品问题是当一个全新的产品加入到推荐系统后，由于缺乏用户对其的评价信息，所以在其加入系统的初期很难被系统推荐。对于新用户问题可以使用用户历史信息以外的其他个人信息（如人口统计信息，性别、年龄、地理位置等）改善推荐效果；或者利用用户在其他系统的信息辅助推荐。对于新产品问题，主要的解决方案是结合协同过滤和内容推荐进行混合推荐[145, 146]。

3) 流行偏置。流行偏置是指推荐系统往往倾向于向用户推荐一些热门的、流行度比较高的产品。简单的进行热门产品的推荐并不能有效体现用户兴趣，导致推荐准确率不高。此外，用户也许已经通过其他途径了解过了这些流行产品（比如畅销书排行榜等），并已对是否购买做出了决定，这就导致推荐结果缺乏新颖性[147]。针对推荐

系统流行偏置问题的研究主要是通过调整推荐结果的流行度分布，在推荐准确率和推荐多样性上进行权衡[27, 147, 148]。

4) 可扩展性。在大数据背景下，推荐系统往往要面对百万级的用户和产品，需要实时的在整个空间上进行相似性查找和排序，并以此进行推荐。这对推荐算法的时间复杂性、空间复杂性以及可并行性提出了严峻的挑战。对于新增数据进行增量学习的方法是解决推荐系统可扩展性的一个有效方式[149]。其中，新增数据包括新用户、新产品以及新评分（亦可是浏览、购买等其它用户行为信息）。

5) 动态性。传统的推荐算法只是简单地利用用户历史信息构造用户-产品评分矩阵进而进行推荐，并未考虑用户信息的时间特征。然而用户的兴趣往往不是一成不变的，文献[150]指出用户未来的兴趣主要受他近期兴趣的影响。此外，用户的部分兴趣还具有周期性或季节性的特征[151]。因此，好的推荐系统需要按照时间和用户状态的不同进行动态推荐。已有一些研究者在利用矩阵分解模型进行推荐时考虑了时间的影响解决推荐结果的动态性问题[151-153]，还有一些研究者利用时间窗来改善推荐结果的实效性[154, 155]。

6) 准确率和多样性。对于推荐系统来说，简单地将流行产品和高评分产品推荐给用户往往就能获得较好的推荐效果。但是，用户可以轻易的从其他途径获取这样的产品信息，也就是说这样的推荐结果对用户的实际信息量并不高。一个好的推荐系统要能为用户发现一些很难被其自发找到，而又很符合用户兴趣的产品，这就需要推荐算法在不大量损失准确率的条件下具有良好的多样性。这方面的主要研究包括：文献[128, 156, 157]设计算法直接提高推荐结果的多样性，文献[27, 39]使用混合推荐算法以期改善推荐结果的多样性，文献[134, 158]通过改善算法对长尾产品的推荐增强推荐结果的多样性，还有一些研究通过推荐新颖的结果来提高推荐算法的多样性[48, 159-162]。

以上问题是协同过滤推荐技术面临的主要问题和挑战。这些问题是制约协同过滤推荐技术进一步发展的瓶颈，也是相关领域研究关注的重点问题。本论文主要针对其中的数据稀疏性和流行偏置的问题进行研究，探索解决这些问题的办法，希望可以通过解决这些问题，提升算法在 Top-*N* 推荐中的准确率和多样性。

1.3 论文主要研究内容和创新点

1.3.1 论文主要研究内容

协同过滤推荐技术涉及多项富有挑战性的任务。针对目前协同过滤推荐中存在的

流行偏置问题和数据稀疏问题，本论文从流行偏置现象、数据稀疏背景下的缺失数据特点和用户行为模式三个方面开展研究工作，针对性的提出了解决方案，包括基于意见的协同过滤推荐算法、缺失数据建模方法及改进型矩阵分解算法、以及两步预测推荐算法，并通过公开的推荐系统数据集对模型和算法的有效性进行验证。具体研究内容论述如下：

1) 针对协同过滤推荐技术的流行偏置问题，即传统推荐算法往往倾向于推荐流行度较高的产品，探索其产生的原因和造成的影响。本论文针对用户行为数据的产生过程对用户模型的影响进行分析，考虑用户行为受到各种广告、口碑、推荐等方面的影响，针对不同行为对用户兴趣模型的表达能力不同，研究利用用户意见信息和产品流行度信息构建用户行为置信度函数的方式，探索使用该置信度函数调节用户行为对用户模型的影响进而改进基于近邻的协同过滤算法的方法。最后，构建实验对置信度函数和改进算法的有效性进行验证。

2) 数据稀疏性是协同过滤推荐技术面临的一个主要问题，也是信息过载问题的一个主要体现。本论文针对数据稀疏背景下的数据缺失原因进行研究，探索利用缺失数据中用户兴趣信息的方式。在数据稀疏背景下，部分缺失数据是由于用户主观选择产生的，这些数据可以用作用户兴趣模型的负例信息。本论文探索对缺失数据建模以获取这些负例信息的方法，并研究利用它们改进矩阵分解推荐算法的方式。最后设计实验验证缺失数据建模方法及改进型矩阵分解推荐算法的有效性。

3) 为提高推荐系统的准确率和多样性，本论文对推荐系统中的用户行为模式进行深入分析，探索如何更真实地描述用户在推荐系统中的行为，提出一种两阶段用户行为模式。针对两阶段用户行为模式，本论文分析了它与传统用户行为模式的区别；并利用真实的推荐系统数据集进行数据分析，验证两阶段用户行为模式的有效性。基于两阶段用户行为模式，探索符合这种行为模式特点的两步预测推荐算法框架，并进一步研究基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。最后，构建实验验证两步预测推荐算法框架和两步预测推荐算法的有效性，其结果也可以进一步验证两阶段用户行为模式的有效性。

1.3.2 论文创新点

本论文围绕 Top-*N* 协同过滤推荐技术开展研究，论文的主要工作和创新点包括：

1) 提出一种基于意见的协同过滤推荐算法。该算法在分析了协同过滤推荐技术中流行偏置问题产生原因的基础上，基于用户对产品偏好的意见信息，使用产品流行

度构建置信度函数，并调整产品在用户模型中的权重，然后依此改进基于用户的协同过滤推荐算法。改进算法可以有效缓解推荐系统的流行偏置问题，并提升推荐结果的准确率和多样性。

2) 提出基于缺失数据建模的改进型 SVD++算法。针对推荐系统的数据稀疏问题，对缺失数据的缺失原因进行研究，考虑缺失数据中包含用户兴趣的负例信息，提出加权法、随机抽样法和近邻抽样法三种对缺失数据建模获取其中负例信息的方法，并通过利用这些负例信息调节推荐模型的训练过程，改进 SVD++推荐算法。实验结果表明改进型 SVD++算法有效提升了算法在 Top-*N* 推荐中的准确率和多样性。

3) 提出两阶段用户行为模式和两步预测推荐算法。针对传统推荐算法的潜在前提假设（即假设用户随机选择产品并进行评分）不满足的问题，提出一种两阶段用户行为模式，将用户选择产品进行评分和用户给出对产品的评分值分割开来。通过对真实的推荐系统数据集的数据分析验证了两阶段用户行为模式的有效性。此外，提出一种以对两阶段用户行为模式进行仿真的方式预测用户行为的两步预测推荐算法框架，并基于该框架，分别提出基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。实验结果表明两步预测推荐算法相对于主流推荐算法有着更好的 Top-*N* 推荐效果，验证了两阶段用户行为模式和两步预测推荐算法的有效性。

1.4 论文的组织结构

论文共分 5 章，各章的主要内容安排如下：

第 1 章，绪论。阐述了论文的研究背景及意义，介绍了国内外在推荐系统领域，尤其是协同过滤推荐技术领域的研究现状，以及评价推荐效果的主要方法，并指出论文的主要研究内容和创新点以及论文的组织结构。

第 2 章，基于意见的协同过滤推荐算法。本章首先介绍了协同过滤推荐中的流行偏置问题，并对其产生原因进行了分析。然后从缓解马太效应影响的角度，以调整不同流行度产品对用户模型的影响为目的，提出两种基于产品流行度的置信度函数。利用此置信度函数改进基于用户的协同过滤推荐算法，依次提出基于用户的加权协同过滤推荐算法、基于共现的协同过滤推荐算法、以及基于意见的协同过滤推荐算法。在实验中，通过与基线算法进行对比，发现基于意见的协同过滤推荐算法可以有效提升算法的 Top-*N* 推荐准确率和多样性。

第 3 章，基于缺失数据建模的改进型 SVD++算法。数据稀疏性是协同过滤技术

面临的主要问题之一，也是信息过载问题的一个主要体现。稀疏的用户-产品评分矩阵中存在着大量的缺失数据，本章对缺失数据的产生原因进行分析，认为部分缺失数据是由于用户主观选择不对产品进行评分而产生的，这部分数据可以视为用户兴趣的负例信息。本章提出加权法、随机抽样法和近邻抽样法三种策略对缺失数据建模以获取用户兴趣的负例信息，并利用获取的负例信息调节推荐模型的训练过程，改进 SVD++算法。实验结果表明改进后的算法比原始的 SVD++算法有着更好的 Top-*N* 推荐准确率和多样性。另外，它们对 SVD++算法的提升效果较使用排序预测方法对 SVD++算法的提升效果更好。

第 4 章，两步预测推荐算法。首先，针对推荐系统中传统用户行为模式的局限性，提出两阶段用户行为模式。其次，详细分析了它与传统用户行为模式的区别，并使用真实的推荐系统数据集对用户评分行为的相关性进行分析，验证了两阶段用户行为模式的有效性。然后在两阶段用户行为模式的基础上，提出一种对该模式的用户行为进行仿真的两步预测推荐算法框架，即第一步预测用户对产品评分的概率，第二步预测用户对产品的评分值。基于该框架，分别提出了利用近邻行为预测用户行为的基于近邻的两步预测推荐算法，以及整合了主题模型和矩阵分解模型的基于模型的两步预测推荐算法。最后，使用 MovieLense 数据集构建实验，对比两步预测推荐算法和主流推荐算法的推荐效果，验证两阶段用户行为模式和两步预测推荐算法的有效性。

第 5 章对本论文的整体工作做了总结分析，指出其中存在的瓶颈和不足，并展望下一步可能的研究方向。

第2章 基于意见的协同过滤推荐算法

2.1 引言

近年来，推荐系统的研究获得了长足的发展。各行各业中的推荐系统帮助人们从浩瀚的信息海洋中寻找有用的信息。作为一个流行的在线购物网站，亚马逊（Amazon）有着非常强大的推荐系统，可以为用户进行个性化的产品推荐。他们首创的“购买此商品的顾客也同时购买”是一个著名的推荐策略[6]。该策略使用基于产品的协同过滤推荐算法，为亚马逊带来了巨大的经济收益。图 2.1 是亚马逊利用该策略进行图书推荐的示意图，用户在浏览《驾驭大数据》的页面时，亚马逊会推荐其他大数据相关书籍给用户，帮助其进行商品选择。

图 2.1 亚马逊图书推荐示意图

基于产品的协同过滤算法利用产品相似度为用户推荐其可能感兴趣的产品，可有效提升购物体验。但是，这种策略存在着一个重大缺陷——“哈利波特问题”[8]：《哈利波特》曾是一本畅销书，小孩会买它，成年人会买它，各个人群都在买它，因此，无论用户正在浏览的商品是什么，他们都有可能会在“购买此商品的顾客也同时购买”栏中看到这本书。也就是说，该推荐策略倾向于向用户推荐那些流行、热门的产品。但是，这样的推荐结果并不一定可以有效体现用户的兴趣。除此之外，用户也许已经通过其他途径了解过这些流行产品（比如畅销书排行榜等），并已做出了是否购买的决定，这使得推荐结果缺乏新颖性[147]。

哈利波特问题不仅仅存在于基于产品的协同过滤推荐算法中，大量推荐算法都一定程度上存在这个问题，该问题也被认为是推荐系统中的流行偏置问题。近年来，已经有很多研究者在试图解决这个问题[21, 27, 147, 148]。

文献[148]提出一种基于产品的协同过滤推荐算法的改进算法，称为 ItemCF-PP 算

[8] http://glinden.bolgspot.com/2006/03/early-amazon-similarities.html

法。他们通过惩罚函数的形式降低流行产品被推荐的可能性，有效提升了算法的推荐效果。

文献[21]使用逆用户频率（Inverse User Frequency，简称 IUF）降低流行产品在基于用户的协同过滤算法中的影响，期望可以缓解推荐系统的流行偏置问题，提高推荐效果。但是，有研究者通过实验说明这种方法（称为 UserCF-IUF）并未对经典的基于用户的协同过滤算法的推荐效果有所改善[163]。这说明简单地使用逆用户频率对产品进行加权并不能有效解决推荐系统的流行偏置问题。

文献[27]使用经典的推荐算法预测用户对未评分产品的评分值，将预测评分值大于预先设定的阈值的产品加入候选产品集合，然后针对不同的推荐目的从候选产品集合中选择不同流行度的产品进行推荐。当追求高的推荐准确率，旨在让推荐结果更符合用户兴趣时，从候选产品集合中选择较流行的产品作为推荐结果；当追求高的推荐多样性，意图让推荐结果更加丰富多彩时，从候选产品集合中选择流行度较低的产品作为推荐结果。该方法是一种利用产品流行度对推荐结果的准确率和多样性进行权衡的策略。它主要通过设定阈值，控制候选产品集合的大小，进而控制权衡的程度。

文献[147]认为不同用户在选择产品时有不同的流行度倾向，并将这一倾向定义为个人流行度倾向（Personal Popularity Tendency），然后通过向用户推荐符合其个人流行度倾向的产品以提高推荐效果。

以上是针对推荐系统流行偏置问题的几种主要解决方法，其中，文献[27, 147, 148]提出的方法以改变推荐结果的流行度分布为主，并没有深入挖掘产生流行偏置问题的原因。本章拟从探索推荐系统产生流行偏置的原因入手解决这一问题，改善推荐效果。下面将分别从流行偏置现象、相关解决方案和推荐算法、实验验证和分析三个方面进行具体介绍。

2.2 流行偏置现象

如 2.1 节中所述，推荐系统往往倾向于向用户推荐流行的、热门的产品，这一现象被称为推荐系统中的流行偏置现象。为了详细的分析这个现象，本节使用真实的推荐系统数据集（Movielens 数据集[9]）进行数据分析。

我们将 Movielens 数据集中的产品按照流行度降序排列，其流行度分布情况如图 2.2 所示。从图中可以发现，不同产品的流行度差别很大，最热门的产品被评分近 600

[9] 使用 Movielens 的 100K 数据集，下载地址为 http://grouplens.org/datasets/movielens/

次，而大量产品被评分次数不足 10 次。图 2.3 绘制了产品流行度的对数曲线，该曲线近似一条直线，说明产品流行度近似服从幂律分布。

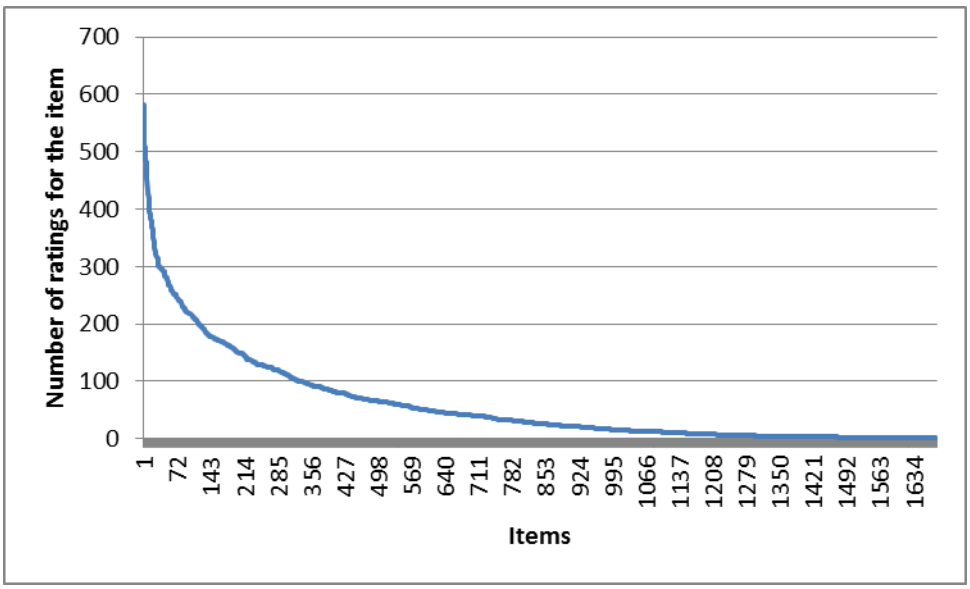

图 2.2 MovieLens 数据集的产品流行度分布

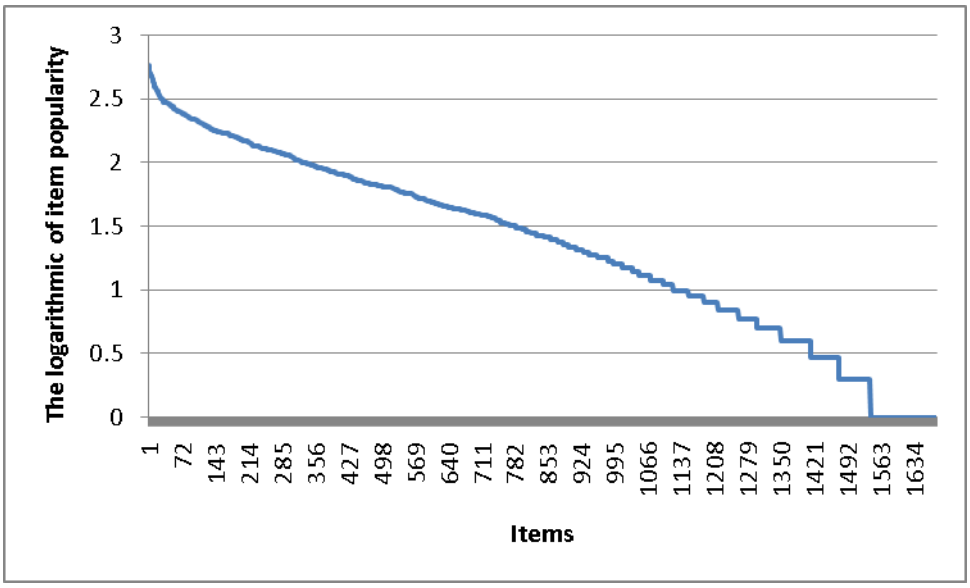

图 2.3 MovieLens 数据集的产品流行度对数曲线

为展示推荐系统的流行偏置现象，我们使用一种经典的基于用户的协同过滤推荐算法为用户进行 Top50 推荐，并统计不同产品的被推荐次数，以观察不同流行度产品的被推荐频率。我们仍然将产品按照流行度进行降序排列，图 2.4 展示了不同产品的被推荐次数。从图中曲线分布可以清晰的看出，流行产品区间的曲线相对更加密集，

也就是说流行的产品比不流行的产品获得了更多的推荐次数。为使结果更加直观，我们计算产品的累计被推荐次数（Accumulative Recommended Times），以便观察产品被推荐次数的增量变化趋势。图 2.5 描绘了产品的累计被推荐次数曲线，图中每个点代表所有流行度大于等于当前产品流行度的产品获得的累计被推荐次数，曲线的斜率代表当前产品的被推荐次数。从图中可以明显的看出，曲线的斜率在流行产品区域内较大，随后逐渐减少。也就是说，产品被推荐的次数随流行度的减少而减少。因此，推荐系统的流行偏置现象可以用马太效应[164]进行解释。

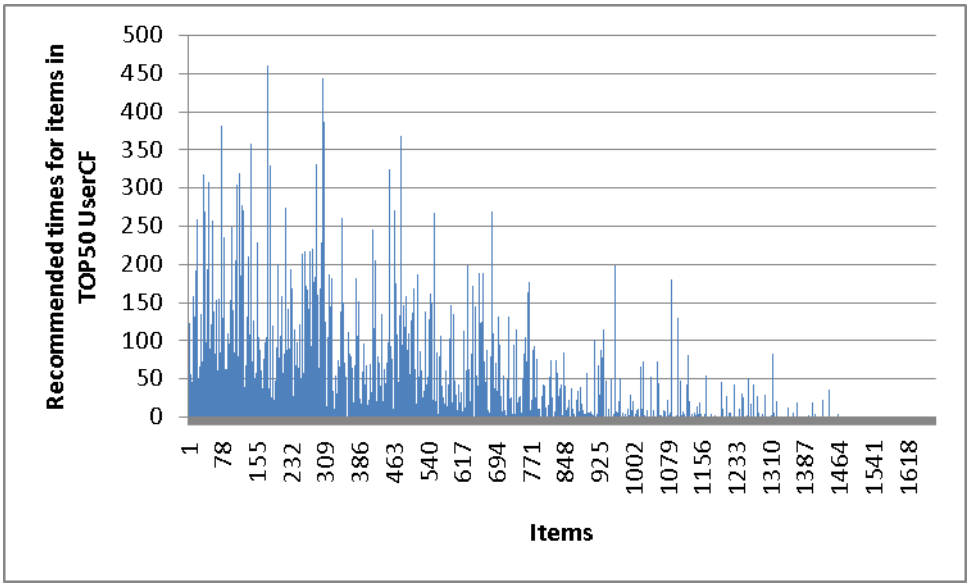

图 2.4 MovieLens 数据集中使用 UserCF 进行 Top50 推荐时产品的被推荐次数

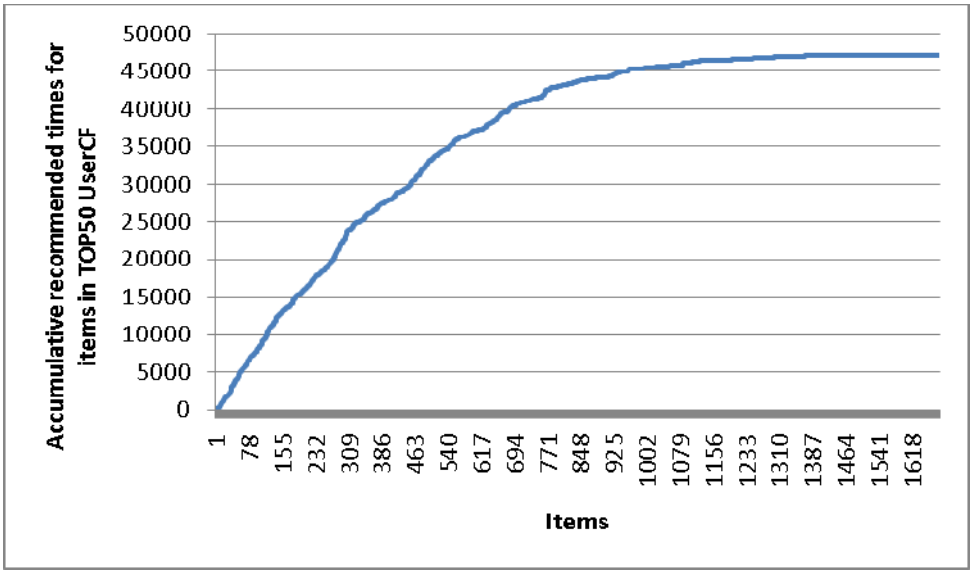

图 2.5 MovieLens 数据集中使用 UserCF 进行 Top50 推荐时产品的累计被推荐次数

马太效应是一种富者愈富，穷者愈穷的现象。该现象广泛地存在于日常生活的方方面面。以社会科学中的马太效应为例，如果一个杰出的科学家和一个平庸的科学家完成了相似的工作，那个杰出科学家的工作往往受到更多的关注和认可，进而会变得越来越杰出[164]。在推荐系统中亦是如此，流行的产品会被更多的推荐，进而被更多的浏览、购买、评分，会变得愈加流行。我们认为这个现象是推荐系统产生流行偏置的主要原因。

相应的，为解决推荐系统的流行偏置问题，可以从缓解推荐系统的马太效应入手进行研究。现有针对推荐系统流行偏置问题的主要研究是从改变推荐结果的流行度分布入手的[27, 147, 148]。这些方法可以一定程度上缓解马太效应，减少流行偏置问题。但它们往往会带来推荐结果准确率的牺牲。与此类解决方案不同，我们旨在从更真实的描述用户行为入手解决流行偏置问题。由于用户行为受到马太效应的影响，用户对不同流行度的产品进行评价的行为对用户兴趣的表达能力是不同的。对于那些频繁出现在畅销榜、广告或者推荐列表中的产品，用户可以较轻易地完成对产品的浏览、购买和评价活动；而对于那些处在长尾[134, 158]处的产品，很少获得广告或者被推荐的机会，用户找到它需要通过主动的检索。相较于前者而言，后者代表用户更大的兴趣。因此，可以认为，用户对产品的兴趣程度与他获取相应产品的难度相关。

换言之，流行偏置现象导致用户获取不同流行度产品的难度不同，进而用户对不同流行度产品的反馈所代表的用户兴趣程度也不同。因此，可以从区别对待不同流行度产品对用户兴趣模型表达能力的置信度入手，研究如何降低马太效应导致的流行偏置现象，进而改进推荐算法，改善推荐效果。下一节将对改进方法进行具体的论述。

2.3 推荐算法

推荐系统中存在着流行偏置问题，用户获取产品的难度受到产品流行度的影响，因此，用户对不同产品产生的反馈信息代表不同程度的用户兴趣。针对以上情况，为有效地描述用户兴趣，我们依据产品流行度信息构建不同产品在用户模型中的置信度函数，并依此改进基于用户的协同过滤推荐算法。本节将分别对置信度函数构建方法和改进后的推荐算法进行介绍。

2.3.1 置信度函数

文献[21]提出使用逆用户频率（IUF）降低流行产品在基于用户的协同过滤算法中

的影响的方法，这即是一种构建置信度函数的方法。对于任意产品 i，其 IUF 置信度可通过公式(2.1)进行计算：

$$w_{IUF}(i) = \log(\frac{|U|}{p_i}) \tag{2.1}$$

其中，U 代表推荐系统中的用户全集，p_i 代表产品 i 的流行度。

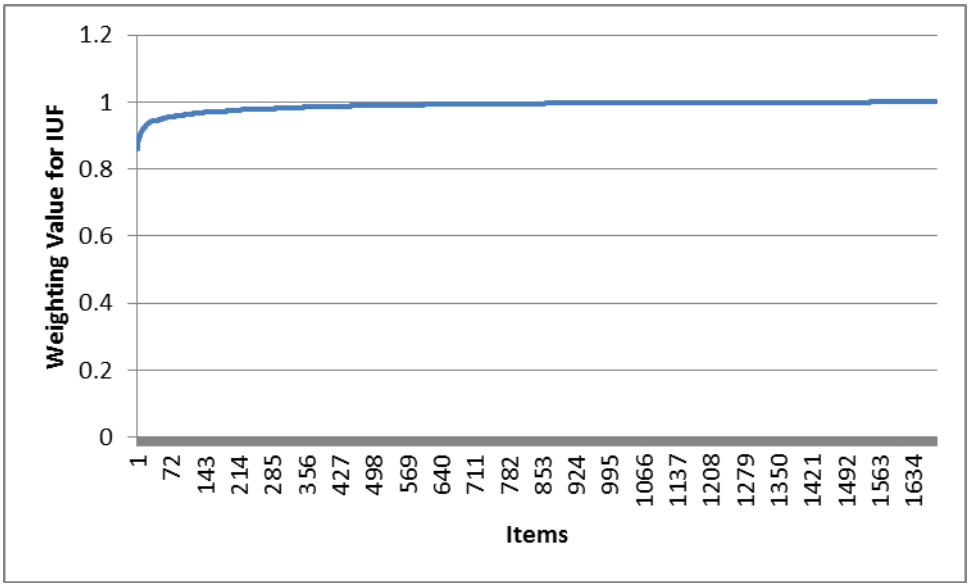

图 2.6 MovieLens 数据集中产品的 IUF 置信度曲线

图 2.6 绘制了 Movielens 数据集中产品的 IUF 置信度曲线。从图中可以发现 IUF 置信度函数的两点主要特征：

- 流行产品的置信度和不流行产品置信度差别不大（不超过 20%）；
- 流行产品的置信度变化速度远大于不流行产品的变化速度。

有研究者证明使用 IUF 置信度函数改善基于用户的协同过滤算法并不能有效提升推荐效果[163]。这也许是由于 IUF 置信度函数所描述的不同产品在用户模型中的影响并不符合真实用户模型的特点。

如 2.2 节中所述，用户对产品的反馈所代表的用户兴趣程度与用户获取相应产品的难易程度相关。产品的被推荐次数是用户获取产品难易程度的一种体现。图 2.5 描绘了产品的累计被推荐次数曲线，图中曲线的斜率变化可以反映出数据集中产品被推荐次数的变化趋势，斜率越大则产品被推荐次数越多。通过观察，发现该曲线具有如下特点：

- 曲线斜率随产品流行度降低而减小，也就是说，产品的被推荐次数和产品流

行度成正比，相应的，用户获取产品的难度与产品流行度成反比；

- 曲线斜率在较流行产品区间内变化很小，也就是说，流行产品的被推荐次数差别不大，用户获取不同的较流行产品的难度差别不大；
- 流行区间和长尾区间的曲线斜率差别很大，也就是说，流行产品和不流行产品的被推荐次数差别很大，用户获取流行产品和不流行产品的难度差别很大。

对比 IUF 置信度函数特点和用户获取产品难易程度特点，可以发现，IUF 置信度函数特点并不能有效表达用户获取产品的难易程度的变化，因此，IUF 函数不适合作为置信度函数用于改善基于用户的协同过滤推荐算法。

为了拟合用户获取产品难易程度的特点，我们提出 Base 置信度函数，该置信度函数计算方法如下：

$$w_{Base}(i)=\frac{1}{\log(p_i)} \tag{2.2}$$

图 2.7 描绘了 Movielens 数据集中产品的 Base 置信度函数曲线。从图中可以发现，Base 置信度函数基本符合了表达用户获取产品难易程度变化的要求，即流行产品之间的置信度差别不大，而流行产品和不流行产品间的置信度差别很大。此外，Base 置信度函数选择对数函数作为基本结构，亦比较符合产品流行度服从幂律分布的特点。

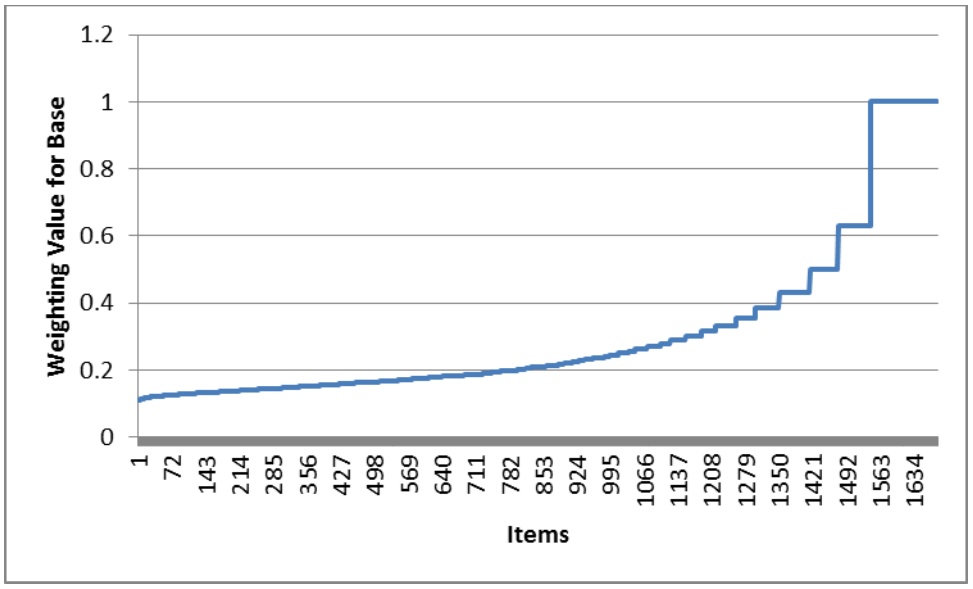

图 2.7 MovieLens 数据集中产品的 Base 置信度曲线

理论上讲，Base 置信度函数已经可以基本满足置信度函数的需求，但是从设计置信度函数的初衷——缓解推荐系统的马太效应角度看，该置信度函数仍然存在改进空间。

在马太效应描述的情景中，少数流行产品因受到广泛的关注而变得越来越流行，而其它大多数产品受到马太效应的影响很小。因此，我们使用一个阈值 T_P，将产品按照流行度分为两个部分，因它们受到不同的马太效应的影响而区别对待。T_P 是 0 到 1 之间的一个小数，其取值代表多大比例的产品将被分到较流行的产品集合中，该集合中的产品最小流行度记为 P_H。剩下的产品被分到另外一个集合中。基于这种集合划分方法，我们构建 Popular 置信度函数。在该置信度函数中，只有那些流行度大于 P_H 的产品，会受到置信度函数的惩罚，而大多数流行度较小的产品将被公平对待。Popular 置信度函数的形式化描述如公式(2.3)所示：

$$w_{Popular}(i)=\begin{cases}\dfrac{1}{\log(p_i)} & ,p_i > P_H \\ 1 & ,p_i \le P_H\end{cases} \tag{2.3}$$

图 2.8 是使用 Movielens 数据集绘制的产品 Popular 置信度函数曲线。观察该曲线，可以发现其特点符合描述用户获取产品难易程度的要求。

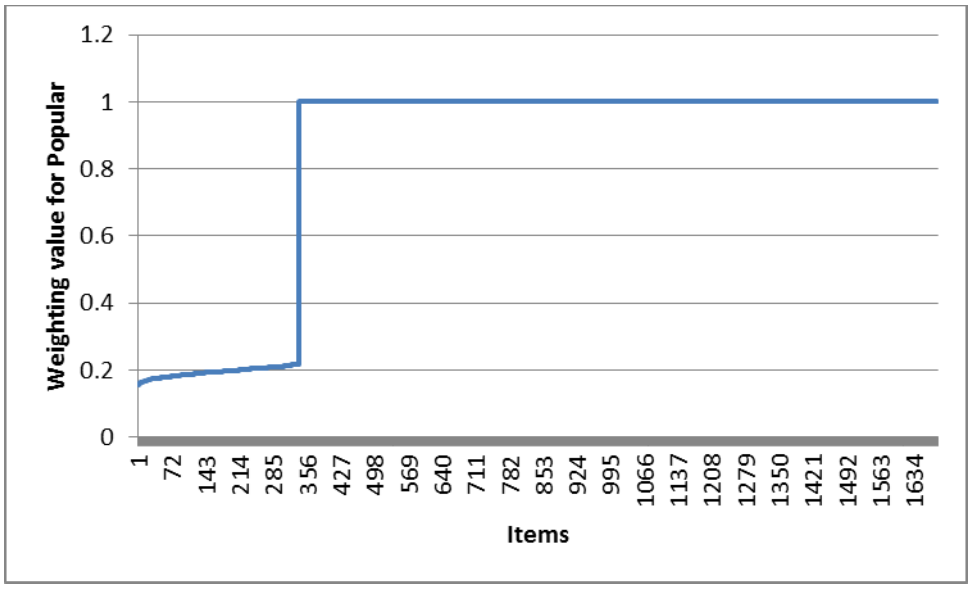

图 2.8 MovieLens 数据集中产品的 Base 置信度曲线

本小节提出了两种基于流行度描述不同产品在用户模型中影响的置信度函数。使用这些置信度函数，可以调整产品在用户模型中的权重，并依此改进基于用户的协同过滤推荐算法。我们提出三种改进模型：基于用户的加权协同过滤推荐算法、基于共现的协同过滤推荐算法和基于意见的协同过滤推荐算法，下面将分别进行介绍。

2.3.2 基于用户的加权协同过滤推荐算法

基于用户的协同过滤推荐算法（UserCF）认为用户会喜欢相似用户喜欢的产品，

并把相应产品推荐给用户[5]。寻找相似用户是该算法的核心部分，计算用户相似度的方法有很多，如 Pearson 相似度，余弦相似度，Jaccard 相似度等。在本章中，UserCF 使用余弦相似度函数作为度量用户相似度的基本方法，具体计算方法可参见公式(1.1)。基于用户相似度，UserCF 为任意用户 u 选择 K 个最相似用户作为他的近邻，并利用近邻对某产品的评分预测用户 u 对该产品的评分。评分预测方法可参见公式(1.7)，在此不再详细介绍。

基于用户的加权协同过滤推荐算法（Weighting-UserCF，简称 WUserCF）是 UserCF 的改进算法，主要改进思想是使用置信度函数改变不同流行度产品对用户模型的表达。具体而言，WUserCF 使用置信度函数调节产品在计算用户相似度时的权重。改进后的用户相似度计算方法如公式(2.4)所示：

$$sim_I(u,a) = \frac{\sum_{i\in I(u,a)} w(i)\cdot r_{ui}\cdot r_{ai}}{\sqrt{\sum_{i\in I(u)}(w(i)\cdot r_{ui})^2}\cdot\sqrt{\sum_{i\in I(a)}(w(i)\cdot r_{ai})^2}} \tag{2.4}$$

其中，$I(u,a)$是用户 u 和用户 a 共同评分的产品集合，$w(i)$是置信度函数，r_{ui}是用户 u 对产品 i 的评分，$I(u)$是用户 u 评分的产品集合。

改进后的相似度计算方法是一种加权的余弦函数，不同产品在用户模型中的影响力不同，影响力的大小通过置信度函数 $w(i)$进行控制。其中，$w(i)$可以是 2.3.1 节中提出的 Base 置信度函数或者 Popular 置信度函数。

基于改进后的相似度函数 $sim_I(u,a)$，WUserCF 进行近邻搜索，然后使用公式(1.7)预测用户对产品的评分值，并依此进行推荐。

2.3.3 基于共现的协同过滤推荐算法

基于用户的加权协同过滤推荐算法使用置信度函数改变不同流行度的产品对用户模型的影响。该算法通过降低流行产品的权重改进基于用户的协同过滤推荐算法中用户相似度的计算方法，这样的改进策略存在部分缺陷。举例来说，对于两个流行产品 i 和 j，假设用户 u 和 v 都对产品 i 有评分行为，这说明他们对产品 i 的兴趣倾向同大多数用户相似，因此，可以认为该产品对计算用户 u 和用户 v 之间的相似度具有较低的置信度；而对于产品 j，只有用户 u 对其有评分行为，用户 v 对其没有评分行为，这可能是由于用户 u 和用户 v 对待产品 j 的兴趣程度是不同的，而且这一区别与多数用户的观点相悖，因此，该产品在区分用户 u 和用户 v 的兴趣时反而应具有较高的置信度。

由此可见，简单的在计算用户相似度时降低所有流行产品的置信度是不可取的，而应考虑用户是否同时对该流行产品有过评分。因此，我们提出一种基于共现的协同过滤推荐算法（Co-rated-based Weighting-UserCF，简称 CWUserCF）。该算法在计算两个用户相似度时，区分产品是否被用户共同评分，对于被用户共同评分的产品，仍使用类似 WUserCF 的处理办法，依据流行度降低相应产品的影响；对于只被一个用户评过分的产品，反而需要加强该产品的影响。因此，我们按照产品是否被两个用户同时评分对置信度函数进行改进，改进后的置信度函数如公式(2.5)所示：

$$w_C(i) = \begin{cases} w(i) & , i \in I(u,a) \\ \dfrac{1}{w(i)} & , i \notin I(u,a) \end{cases} \tag{2.5}$$

其中，*w(i)*可以使用 2.3.1 节中介绍的原始置信度函数（Base 或者 Popular 皆可）。基于该置信度函数，CWUserCF 使用公式(2.4)计算用户相似度[10]，使用公式(1.7)预测用户对产品的评分，并依此进行推荐。

2.3.4 基于意见的协同过滤推荐算法

基于共现的协同过滤推荐算法在基于用户的加权协同过滤推荐算法的基础上考虑了产品在不同用户行为中的共现情况，并对不同情况使用不同置信度计算方法。理论上讲，CWUserCF 较 WUserCF 更贴近用户行为特征，但是，该方法仍存在部分缺陷，即没有考虑用户对产品的意见信息。为此，我们引入用户对产品的意见信息，提出基于意见的协同过滤推荐算法（Opinion-based Weighting-UserCF，简称 OWUserCF）。该算法使用类似 CWUserCF 的思想，对于用户具有相似意见的产品，依据流行度降低其对用户模型的影响；对于用户具有不同意见的产品，反而依据流行度提升相应产品对用户模型的影响。

在 OWUserCF 中，我们将用户 *u* 和用户 *a* 具有相似意见的产品集合定义为 *S(u,a)*，该集合中的任意产品 *i* 需要满足如公式(2.6)所示条件：

$$(r_{ui} - \overline{r(u)}) \cdot (r_{ai} - \overline{r(a)}) > 0 \tag{2.6}$$

其中，$\overline{r(u)}$ 是用户 *u* 对已评分产品的平均评分。平均评分用于区分用户对某产品的评分是正意见（大于平均评分）还是负意见（小于平均评分）。两个用户具有相似意见的产品是他们同时具有正意见或负意见的产品。基于产品集合 *S(u,a)*，OWUserCF 对

[10]需将公式(2.4)中的 *w(i)*替换为公式(2.5)中的 $w_c(i)$。

置信度函数进行调整，调整后的置信度函数计算方法如公式(2.7)所示：

$$w_O(i) = \begin{cases} w(i) & , i \in S(u,a) \\ \dfrac{1}{w(i)} & , i \notin S(u,a) \end{cases} \tag{2.7}$$

与公式(2.5)类似，公式(2.7)中的 *w(i)*亦为 2.3.1 节中提到的原始置信度函数。此外，由于在置信度函数中考虑了用户的意见信息，相应的，在计算用户相似度时亦需做出对应调整。公式(2.8)是考虑了用户意见信息的相似度计算方法。OWUserCF 基于该式计算用户相似度，然后使用公式(1.7)预测用户评分，生成推荐结果。

$$sim_O(u,a) = \frac{\sum_{i \in I(u,a)} w_O(i) \cdot (r_{ui} - \overline{r(u)}) \cdot (r_{ai} - \overline{r(a)})}{\sqrt{\sum_{i \in I(u)} (w_o(i) \cdot (r_{ui} - \overline{r(u)}))^2} \cdot \sqrt{\sum_{i \in I(a)} (w_O(i) \cdot (r_{ai} - \overline{r(a)}))^2}} \tag{2.8}$$

综上所述，本节提出 Base 和 Popular 两种置信度函数，用于调节不同流行度产品对用户兴趣模型的影响。在这两种置信度函数的基础上分别提出基于用户的加权协同过滤推荐算法、基于共现的协同过滤推荐算法、以及基于意见的协同过滤推荐算法。下面我们将构建实验，分析这两种置信度函数和三种推荐算法的有效性。

2.4 实验与分析

2.4.1 实验设置

本章利用 MovieLens 的 100K 数据集进行实验验证。这个数据集包含了 943 个用户对 1682 部电影的 100000 条评分记录。其中，每个用户至少对 20 部电影作出评分。该数据集中的评分值是 1-5 之间的整数，5 表示最喜欢。我们将整个数据集随机分为 5 份，每次实验使用其中 4 份作为训练集，推荐算法使用这部分数据生成推荐结果；另外 1 份作为测试集，验证推荐效果。实验共进行 5 次交叉验证。

为了评价推荐算法的有效性，本章使用 1.2.4 节中介绍的 NDCG 和 1-Call 评价方法对算法的推荐准确率进行评价，使用 COV 和 CIL 评价方法对算法的多样性和新颖性进行评价，并且选择 NDCG+进行对照，评价推荐算法的排序预测能力。

实验将对比本章提出的三种算法（WUserCF、CWUserCF 和 OWUserCF）分别使用 Base 置信度函数和 Popular 置信度函数的推荐效果，并将其与 UserCF、ItemCF、UserCF-IUF 和 ItemCF-PP 四种基线算法的推荐效果进行对比分析。UserCF[5, 88]和 ItemCF[6]是经典的基于近邻的协同过滤算法，其中 UserCF 是本章算法改进的源算法；

UserCF-IUF 和 ItemCF-PP 分别是文献[21]和文献[148]中提出的解决流行偏置问题的方法。对于以上算法我们分别计算它们在 Top1、Top3 和 Top5 推荐中的 NDCG 和 NDCG+ 的效果，以及它们在 Top5 推荐中的 1-Call、COV 和 CIL 效果。

这些算法都是基于近邻的协同过滤算法，近邻数是这类算法的一个重要参数。本章中，以上算法都使用 50 个近邻进行相关计算，该值是基于近邻的协同过滤算法的一种典型参数值。此外，Popular 置信度函数中需设定一个阈值参数 T_P，我们将其设为 0.1，并将在实验中讨论不同 T_P 对推荐效果的影响。

2.4.2 实验结果

本章实验对十种算法的推荐效果进行分析、比较。其中，四种算法是已有基线算法，六种是本章提出的算法。表 2.1 中列出了这些算法推荐效果的对比结果。表中的每一行是一种算法，每一列是一种评价方法。对于每一种评价方法，加粗了该评价方法下表现最好的三种算法。下面将分别针对不同的评价方法分析各算法的表现。

表 2.1 不同算法的推荐效果比较

Algorithms		NDCG			1-Call	COV	CIL	NDCG+		
		1	3	5				1	3	5
Benchmark	UserCF	**0.0089**	**0.0128**	**0.0148**	**0.09**	110	66	**0.74**	**0.73**	**0.73**
	ItemCF	0.0005	0.0006	0.0007	0.01	**156**	**142**	0.56	0.61	0.65
	UserCF-IUF	0.0068	0.0097	0.0132	**0.09**	95	58	**0.74**	**0.73**	**0.73**
	ItemCF-PP	0.0005	0.0003	0.0006	0.01	**142**	**139**	0.38	0.39	0.40
Proposal	WUserCF-Base	0.0048	0.0087	0.0128	**0.09**	97	57	0.73	0.72	0.72
	WUserCF-Popular	0.0023	0.0096	0.0103	0.07	110	67	0.70	0.69	0.70
	CWUserCF-Base	0.0063	0.0087	0.0126	0.08	88	52	0.73	**0.73**	0.72
	CWUserCF-Popular	0.0040	0.0064	0.0082	0.06	98	61	0.73	0.71	0.71
	OWUserCF-Base	**0.0190**	**0.0264**	**0.0286**	**0.16**	115	62	**0.77**	**0.77**	**0.77**
	OWUserCF-Popular	**0.0323**	**0.0353**	**0.0370**	**0.16**	**122**	**70**	0.72	**0.74**	**0.75**

表中第 3、4、5 列是不同算法在 Top1、Top3 和 Top5 推荐中的 NDCG 效果。其中，本章提出的 OWUserCF-Popular 和 OWUserCF-Base 表现最好，其次是 UserCF，该算法是基线算法中效果最好的方法。对比这三种方法的表现，可以看出即使是 OWUserCF 中相对较差的 OWUserCF-Base 方法，其 NDCG 表现也较 UserCF 提高了大约 100%。由此可以看出，在计算用户相似度时，考虑用户对产品的意见信息，基于产品的流行度，调整产品在用户模型中的置信度，可以有效提升基于用户的协同过滤算法的推荐准确率。但是，WUserCF 和 CWUserCF 并没有按照预期提升推荐准确

率。反而，这两类算法的推荐准确率都较传统的 UserCF 有了一定下降。这说明简单的按照流行度对产品进行加权，或者仅仅考虑用户的共同评分行为并不能客观描述马太效应对用户行为的影响，其错误地调整产品在用户模型中的置信度反而降低了模型对用户兴趣描述的准确度，因此降低了算法的推荐准确率。此外，UserCF-IUF 的推荐准确率也较 UserCF 小，这进一步验证了文献[163]中提到的使用 IUF 置信度并不能有效提升推荐效果的观点。考虑不同置信度函数的影响，OWUserCF-Popular 较 OWUserCF-Base 有着更高的推荐准确率。通过计算可知，使用 Popular 置信度函数可以进一步提升 OWUserCF-Base 的 NDCG 表现，提升效果可达到 29%以上。这说明 Popular 是比 Base 更有效的置信度函数，也验证了 2.3.1 节中提出的只有少数较流行的产品受马太效应影响比较大的观点。

表中第 6 列是各算法在 Top5 推荐中的 1-Call 表现。与 NDCG 相似，同样是 OWUserCF 表现最好，其相较 UserCF 的提升在 73%以上。这一现象从保证 Top-*N* 推荐中至少有一个相关产品的能力的角度验证了基于意见的协同过滤推荐算法的有效性。

表中第 7、8 列分别是不同算法的 COV 和 CIL 表现。其中，ItemCF 和 ItemCF-PP 表现最好。但是这两个算法的推荐准确率是所有算法中最差的。它们靠牺牲准确率而获得了较高的推荐多样性和新颖性。如果不考虑这两个算法，OWUserCF 在剩余算法中表现最好，说明基于意见的协同过滤推荐算法在保证了推荐准确率的前提下，还有效提升了推荐结果的多样性和新颖性。此外，OWUserCF 良好的覆盖率表现表明它相较 UserCF 而言可以向用户推荐更多的产品，有效缓解了流行偏置问题。

表中第 9、10、11 列是不同算法的 NDCG+表现。在该评价方法中，OWUserCF 仍然获得了最好的表现，说明基于意见的协同过滤推荐算法可以很好地预测用户对产品的排序结果，将用户评分高的产品排在推荐结果的较前位置。

总而言之，基于意见的协同过滤推荐算法通过利用考虑用户意见倾向及产品流行度信息的置信度函数，调整不同产品在用户模型中的权重，有效提高了基于用户的协同过滤算法的推荐效果。

此外，OWUserCF 使用 Popular 置信度函数能获得比使用 Base 置信度函数更好的 Top-*N* 推荐表现。在 Popular 置信度函数中，有一个重要的参数 T_P，产品按照 T_P 的取值被划分为两个不同的子集，并使用完全不同的置信度策略。因此，T_P 的不同取值会影响到子集的划分方式，进而影响 OWUserCF 的推荐效果。下面我们将构建实验，分

析不同的 T_P 取值对 OWUserCF 推荐效果的影响。在该实验中，T_P 的取值从 0 增长到 1，以 0.1 为增长的步长。

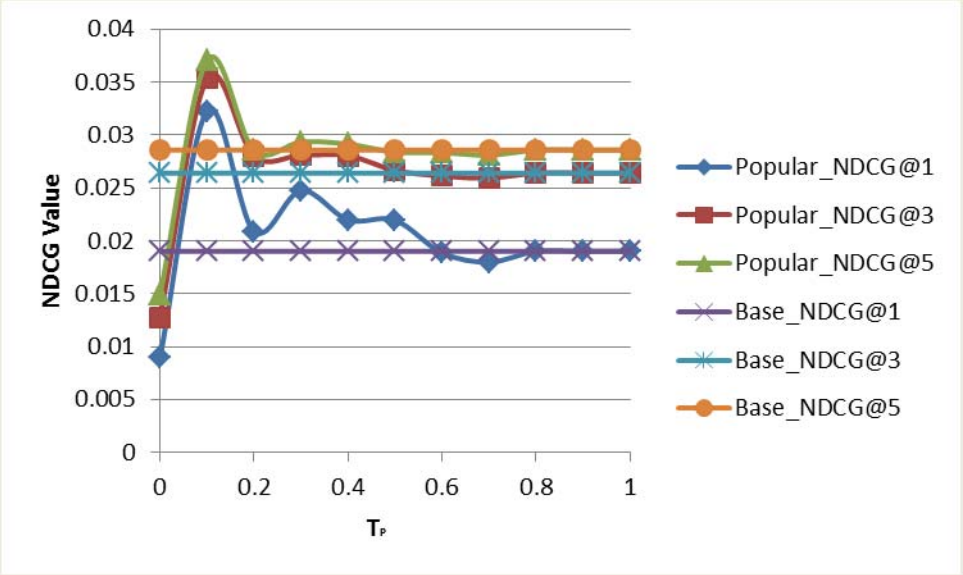

图 2.9 OWUserCF 的 NDCG 表现随 T_P 变化的曲线

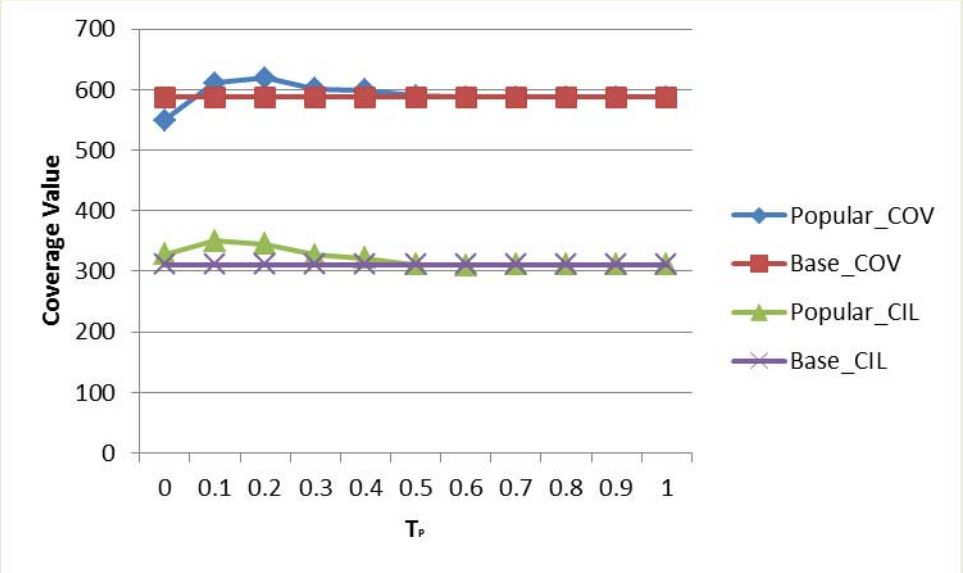

图 2.10 OWUserCF 的多样性表现随 T_P 变化的曲线

图 2.9 和图 2.10 分别展示了 OWUserCF 使用不同置信度函数的 NDCG、COV 和 CIL 表现。当 T_P 为 0 时，OWUserCF-Popular 表现最差，甚至比使用 Base 置信度函数的方法表现都差。这主要是由于当 T_P 取 0 值时，任何产品的 Popular 置信度函数取值都是 1，此时，OWUserCF 退化为传统的 UserCF。当 T_P 取值为 0.1 到 0.5 之间时，OWUserCF-Popular 推荐效果优于 OWUserCF-Base。其中，当 T_P 取值为 0.1 时，OWUserCF-Popular 获得了最佳表现。当 T_P 大于 0.5 时，OWUserCF-Popular 和

OWUserCF-Base 的表现基本相同。这主要是因为流行度后 50%的产品被评分次数只占总评分次数的 7%左右。从用户群体的角度看，调整该部分产品的权重对用户模型的影响很小。

综上所述，基于意见的协同过滤推荐算法提升了基线算法的 Top-*N* 推荐效果，推荐结果对长尾产品的覆盖率更高，有效缓解了流行偏置问题。另外，本章提出了两种置信度函数，其中，Popular 置信度函数在 T_P 取值为 0.1 到 0.5 之间时表现优于 Base 置信度函数，验证了只有少数热门产品受马太效应影响较大的观点。

2.5 本章小结

“哈利波特”问题是推荐系统研究领域的一个经典问题。该问题的本质是推荐系统中的流行偏置问题。本章通过对流行偏置现象的分析研究，发现用户在推荐系统中的行为受到了马太效应的影响，然后从缓解马太效应入手，提出使用置信度函数调节不同流行度产品在用户模型中权重的方式，对基于用户的协同过滤算法进行改进，提升算法的 Top-*N* 推荐效果。

本章提出了 Base 和 Popular 两种置信度函数，其中 Base 函数对所有产品按照流行度使用统一方式设置置信度，Popular 函数通过一个预定义的阈值 T_P，将产品划分为两个集合，只对较流行的产品调整置信度。基于这两种置信度函数，本章分别提出了基于用户的加权协同过滤推荐算法、基于共现的协同过滤推荐算法和基于意见的协同过滤推荐算法。其中，基于意见的协同过滤算法在计算用户相似度时考虑了两个用户是否对同一产品具有相似意见的信息。该算法使用置信度函数降低用户具有相似意见的产品对用户模型的影响，提升不同意见的产品对用户模型的影响。实验结果表明，基于意见的协同过滤推荐算法可以有效提升基线方法在 Top-*N* 推荐中的准确率、多样性和新颖性，有效缓解了推荐系统的流行偏置问题。

第3章 基于缺失数据建模的改进型 SVD++算法

3.1 引言

数据稀疏是推荐系统面临的一个主要问题，也是信息过载问题的一个重要体现。经典的协同过滤推荐算法使用用户对产品的评分矩阵训练推荐模型，预测用户行为，并生成推荐结果。在这样的背景下，数据稀疏性主要体现在用户-产品评分矩阵是一个稀疏矩阵，即每个用户只对少数产品有评分行为，每个产品只有少数用户对它有过评分。举例来说，Netflix 数据集中包含 480189 位用户对 17770 个产品的，共计 100480507 条评分数据，计算可知，已评分数据约占可评分数据的 1%，有大约 99%的数据是缺失数据；Movielens 的 100K 数据集中包含 943 位用户对 1682 部电影的，共计 100000 条评分数据，已评分数据约占可评分数据的 6%，大约 94%的数据是缺失的。

大量数据的缺失，为选择和设计推荐算法带来了极大的困难。已有推荐算法大多基于实际的观测数据，以评分预测准确率（如 RMSE）或排序预测准确率（如 NDCG）为学习目标，训练模型并计算推荐结果。这类推荐算法在数据全集上的有效性，往往基于一个潜在的假设条件：即数据的缺失是一种随机的缺失。如果该假设成立，那么仅依据观测数据训练出的模型可以很好的描述数据全集的特征，并对数据全集做出无偏估计。但是，已有研究表明，数据的缺失并不是一种随机缺失[165-167]。因此，仅使用观测数据训练出的推荐模型并不能无偏地描述用户和产品的特征，相应的推荐算法存在改进的空间。

文献[165]对数据的缺失方式进行了研究，他们使用 Yahoo!LaunchCast 数据集进行调研，以两种方式让用户对音乐进行评分。其中，一种是用户自主选择音乐，并进行评分的 Yahoo!LaunchCast 平台数据集；另一种是系统随机选择音乐让用户进行评分的 Yahoo!LaunchCast 调研数据集。前者类似于实际推荐系统中的用户评分行为，后者则是对数据全集的抽样。图 3.1 和图 3.2 分别展示了两种评分方式的统计结果。从图中可以明显的发现，低评分数据在调研数据集中所占的比例要远大于在平台数据集中的比例。也就是说，低评分数据相对于高评分数据有更高的缺失概率。文献[166]在该调研的基础上，提出一种假说，认为高评分产品是和用户兴趣相关的产品，用户对这类产品的评分是随机缺失的；另外，与相关产品相比，用户对非相关产品的评分具有更

大的缺失概率。基于这一假说，作者对缺失数据建模，并依此改进矩阵分解模型，使其获得了更好的 Top-*N* 推荐效果。

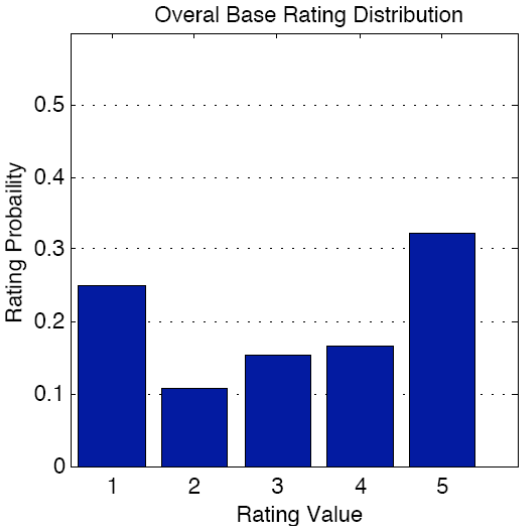

图 3.1 Yahoo!LaunchCast 平台数据集评分分布图[165]

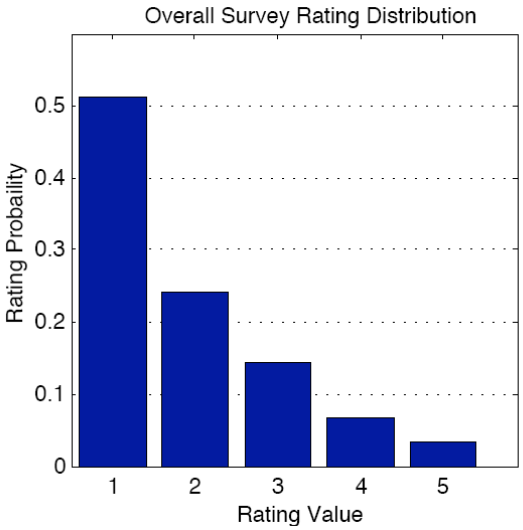

图 3.2 Yahoo!LaunchCast 调研数据评分分布图[165]

与以上观点类似，我们也认可推荐系统中数据的缺失不是一种随机缺失。但我们的观点又与他们有所不同。文献[165-167]认为缺失数据中主要是低评分数据（Negative Ratings），而我们认为缺失数据中主要是用户不愿意去评分的数据，这些数据可以作为用户兴趣的负例信息（Negative Examples）。

在典型的推荐系统应用场景中，用户可以自主的选择产品，并依据喜好程度对其进行评分。例如，在电子商务平台，用户往往会对其已经购买的产品进行评分；在电

影网站，用户会对其已经看过的影片给出评论和评分。而对于网站中大量用户不关注、不感兴趣的商品或者电影，用户很少会刻意地购买或观看后给出一个评分。因此，可以认为这些缺失数据是用户主观选择不对其进行评分的。

本章基于这个观点，对于任意用户，将所有产品划分为两个集合。其中，一个集合是该用户想要进行评分的产品集合，它们是用户兴趣的正例信息，这个集合被记做（c^+）；另一个集合是该用户不想评分的产品集合，它们是用户兴趣的负例信息，这个集合被记做（c^-）。在推荐系统中，观测数据是 c^+的一部分，c^+的其它部分和 c^-共同构成缺失数据。推荐算法的目标是识别缺失数据中属于 c^+的部分，并将其推荐给用户。但是完成这个目标存在着一个很大的困难，即如果将这个目标视为一个二分类的问题，只能从观测数据中获取正例信息，缺乏显性的负例信息，无法有效训练模型。幸运的是，这些负例信息并非不存在，而是隐藏于缺失数据中。如果可以有效地从缺失数据中识别出负例信息，再利用这些信息与正例信息一起训练推荐模型，就可找到缺失的正例信息，并将其推荐给用户。

针对以上问题，本章提出加权法、随机抽样法和近邻抽样法三种对缺失数据建模，获取其中负例信息的方法。方法的细节将在下一节进行具体介绍。这是三种通用的方法，可以不同程度上获取缺失数据中的负例信息。它们可以用来改进各种协同过滤推荐算法。本章将利用它们改进 SVD++推荐算法，并使用 MovieLens 数据集，验证这些缺失数据建模方法和改进后的推荐算法的有效性。

3.2 缺失数据建模方法

在推荐系统中，用户可以自由选择产品，并对其进行评分。文献[165]对 Yahoo!Lanchcast 数据集的调研结果显示，有 93.9%的用户会经常对他们喜欢（Love）的产品进行评分，而只有 36.5%的用户选择会经常对他们觉得中立（Neutral）的产品进行评分。该调研的背景是 Yahoo 的音乐网站，待评分对象是以歌曲为主的音乐类型产品。这类产品往往免费收听，时长大多在 3-5 分钟之间，具有免费且时间花费小的特点，因此，用户收听、评分付出的成本也相对较小。即便如此，也只有少部分用户会对他们感觉中立的产品评分。如果将场景切换到电子商务、电影等有经济花费或者时间成本比较大的背景下，将会有更小比例的用户会对他们感觉中立的产品进行评分。由此可以推测，推荐系统观测到的用户评分行为是用户选择倾向的体现，是用户兴趣的体现。如 3.1 节中所述，针对任意用户，我们将其感兴趣、并想评分的产品划

分到集合 c^+中；将用户不感兴趣或觉得中立的、不愿意评分的产品划分到集合 c^-中。c^+中的数据是用户兴趣的正例信息（Positive Examples），c^-中的数据是用户兴趣的负例信息（Negative Examples）。观测数据是正例信息的一部分，正例信息中的缺失部分和负例信息共同组成了缺失数据。

推荐系统的目标是从缺失数据中识别出未评分的正例信息，然后将这部分正例信息中用户可能会评高分的产品推荐给用户。但是，由于观测数据中只有正例信息，推荐模型很难只利用观测数据对正例信息和负例信息进行划分。因此，本章提出使用加权或抽样的方式对缺失数据建模并获取其中的负例信息，以帮助推荐模型对正、负例信息进行区分。为了在一个统一的推荐模型内对两类信息进行描述，本章参照正例信息描述方式，为负例信息设置默认评分值（r_m）。由于正、负例信息是两个不同的集合，因此它们相应的评分取值亦应该在两个不同的范围。典型的推荐系统评分取值范围为 1-5 或者 1-10，因此，本章选择 r_m 的默认取值为 0。这样的取值既可以将正负例信息表示在一个统一的值域中（0-5 或者 0-10），又可以保证正负例信息可以区分开来（正例信息大于 0，负例信息等于 0）。3.4 节中将通过实验分析 r_m 的不同取值对推荐效果的影响。

3.2.1 加权法

加权法（Weighting Scheme）认为缺失数据一定程度上都是负例信息。但是这些负例信息并不是用户显性提供的。因此，相对于观测数据是正例信息而言，缺失数据是负例信息的置信度较低。加权法通过调节不同数据在推荐模型中的权重描述这种置信度的区别，其权重计算方法如公式(3.1)所示：

$$w_{ui}=\begin{cases}1, & <u,i>\in R\\ \delta, & <u,i>\notin R\end{cases} \tag{3.1}$$

其中，R 是观测数据集合，包含所有已知的用户对产品的评分情况；w_{ui} 是用户 u 对产品 i 的评分数据的权重。如果<u,i>是观测数据，则其权重值为 1，反之，如果属于缺失数据，则其权重为 δ。δ 是预定义的为缺失数据设定的全局权重值。

加权法使用 r_m 作为缺失数据的默认评分值，以便在一个统一的模型中同时描述观测数据和缺失数据。因此，加权法可以将数据全集表示为公式(3.2)所示的形式：

$$r_{ui}^{*}=\begin{cases}r_{ui}, & <u,i>\in R\\ r_m, & <u,i>\notin R\end{cases} \tag{3.2}$$

其中，r_{ui}^{*} 是统一模型中的数据元素。当<*u,i*>属于观测数据时，其值为已知评分值 r_{ui}；当<*u,i*>是缺失数据时，其值为缺失数据的默认评分值 r_m。

使用加权法后，推荐算法的优化目标变为最小化加权 Frobenius 损失函数。该损失函数可以形式化的表示为：

$$\mathcal{L}(R^{*}) = \sum_{<u,i>} w_{ui} \cdot (r_{ui}^{*} - \hat{r}(u,i))^2 \tag{3.3}$$

其中，R^{*}是加权法构建的统一数据模型，它是统一模型中所有数据（r_{ui}^{*}）的集合；$\hat{r}(u,i)$ 是推荐算法预测的用户 u 对产品 i 的评分值。加权法可以与传统的协同过滤推荐算法进行结合，使用加权法利用缺失数据中的负例信息，基于公式(3.3)调整推荐算法的训练过程，进而改善推荐效果。

加权法是一种简单的缺失数据建模方法，在一些已有研究中，已经提出过类似的解决方案[115, 166]。例如，文献[166]提出一种称为 AllRank-Regression 的算法。该算法为缺失数据设置默认评分值，并用其与观测数据一起训练推荐模型。广义上说，该算法与加权法可以看作是一个模型。它们的主要区别是处理缺失数据的理念不同，AllRank-Regression 认为缺失数据包含大量低评分数据，而加权法认为缺失数据包含大量用户没有评分意愿的负例信息。文献[115]中提出的 PureSVD 是另一种与加权法类似的利用缺失数据的推荐算法。该算法简单的将所有缺失数据都视作用户对产品评 0 分。该方法是加权法的一个特例，即 r_m=0，δ=1 的加权法。我们将在 3.4 节的实验中将这两种算法作为基线算法，对其推荐效果进行对比分析，以论证不同的缺失数据建模方法对推荐效果的影响。

3.2.2 随机抽样法

加权法认为缺失数据一定程度上都是负例信息。这一假设在大多数情况下都是成立的，后文的实验也会验证加权法可以有效提升算法的推荐效果。但是，推荐算法原本只需针对观测数据优化模型，使用加权法后则需要对数据的全集进行优化，导致模型训练时的计算复杂度大幅提升。为缓解这个问题，我们提出一种利用随机抽样法对缺失数据建模并获取其中负例信息的方法。该方法较加权法而言，可有效降低使用缺失数据的计算复杂度。

抽样法（Sampling Scheme）认为缺失数据中部分数据是负例信息。随机抽样法（Random Sampling Scheme）是抽样法的一种，它利用随机算法从缺失数据中抽取一定比例（θ）的数据作为负例信息。抽取出的数据集合记为 *RandomN*。该集合与观测

数据一起构成随机抽样法的统一数据模型 $R^* = R \cup RandomN$，其中的数据元素可以形式化的描述为：

$$r_{ui}^* = \begin{cases} r_{ui}, & <u,i> \in R \\ r_m, & <u,i> \in RandomN \end{cases} \tag{3.4}$$

随机抽样法需要使用 R^*中的数据对推荐模型进行训练和优化。其中，观测数据 R 的大小已经固定，因此，随机抽样法的计算复杂度取决于 $RandomN$ 的大小。当 θ 取值为 1 时，$RandomN$ 即为缺失数据的全集，随机抽样法的计算复杂度与加权法相当；当 θ 取值为 0 时，$RandomN$ 为空集，随机抽样法与不利用缺失数据的原始推荐算法计算复杂度相当；当 θ 取值在 0-1 之间时，θ 越小，算法计算复杂度越小。在 3.4 节的实验中，θ 取值为 0.2 时，随机抽样法的推荐效果最佳。这说明了相对加权法而言，随机抽样法大幅降低了训练推荐模型的计算复杂度。

由于随机抽样法相当于是从缺失数据中选择部分数据，将其加入观测数据共同影响推荐模型的训练。该方法对推荐算法的改变仅仅在于训练所用数据源的改变，并不会对推荐模型本身造成影响。因此，使用随机抽样法后，推荐算法的优化目标与源方法保持一直，仍为最小化 Frobenius 损失函数。该损失函数的形式化描述如公式(3.5)所示：

$$\mathcal{L}(R^*) = \sum_{<u,i>} (r_{ui}^* - \hat{r}(u,i))^2 \tag{3.5}$$

值得注意的是，随机抽样法使用随机算法从缺失数据中获取负例信息。因此，为降低使用随机抽样法带来的不稳定性，需要在训练推荐模型的每一次迭代中都重新进行抽样。

3.2.3 近邻抽样法

随机抽样法只使用部分缺失数据，可以有效降低使用缺失数据的计算复杂度。但是，随机抽样法随机地从缺失数据中抽样获得负例信息，使得缺失的正例信息与负例信息有相等的被抽样的概率。为降低缺失正例信息被抽样的概率，提高负例信息被抽样的概率，我们提出一种近邻抽样法（Neighbor-based Sampling Scheme），该方法利用近邻信息启发式地从缺失数据中对负例信息进行抽样。

与基于近邻的协同过滤推荐算法类似，近邻抽样法认为，如果用户的近邻都未对某产品评分，那么当前用户就有很大的可能不对该产品评分。因此，对于任意用户，

近邻抽样法首先计算用户相似度[11]，并寻找当前用户的近邻用户；其次，从当前用户未评分的产品集合中获取近邻亦未评分的产品子集，将该集合作为负例信息的候选数据集合；最后，从候选数据集合中抽取一定比例（θ）的产品作为当前用户的负例信息。近邻抽样法的形式化描述如图 3.3（Algorithm1）所示：

Algorithm 1. The neighbor-based sampling scheme.

Input:
The rating matrix R, the random ratio θ, the neighbor size k.
Output:
The neighbor-based sampling matrix *NeighborN*.

1: **for** each user $u \in U$ **do**
2: Find *N(u)*: the top-*K* most similar users of user *u*
3: Find *OUT(u)*: the item set, in which items have not been rated by user *u*
4: Find *C(u)*: the candidate item set, a sub set of *OUT(u)*, in which items have not been rated by all users in *N(u)*
5: Random select θ percentage of items in *C(u)* into *NeighborN*
6: **end for**

图 3.3 近邻抽样法伪代码

通过该算法，我们可以获取抽样所得的负例信息集合（*NeighborN*），该集合与观测数据一起构成近邻抽样法的统一数据模型 $R^* = R \cup NeighborN$，该模型中的元素可以形式化的描述为：

$$r_{ui}^* = \begin{cases} r_{ui}, & <u,i> \in R \\ r_m, & <u,i> \in NeighborN \end{cases} \tag{3.6}$$

基于统一数据模型 R^*，近邻抽样法可以利用公式(3.5)以最小化 Frobenius 损失函数为优化目标，训练推荐模型。

3.3 推荐算法

3.2 节介绍了通过加权或抽样对缺失数据进行建模的方法。这些方法可以与各种协同过滤推荐算法结合，辅助解决数据稀疏性的问题，进而改善推荐效果。本节以

[11] 近邻抽样法利用近邻信息启发式的寻找用户不愿评分的产品集合。该方法希望找到的相似用户是有共同的（不）评分习惯的用户。因此，使用用户评分行为计算用户相似度的 Relevance 方法较适合这样的需求。Jaccard 相似度是一种 Relevance 相似度，本章选择它作为近邻抽样法的相似度函数，其具体计算法方法可参见公式(1.3)。

SVD++算法为原始推荐算法，使用 3.2 节中介绍的缺失数据建模方法对其进行改进，以期提升算法的 Top-*N* 推荐效果。

SVD++算法[35]是推荐系统领域针对评分预测问题的最优算法之一。该算法认为用户的隐性反馈信息是非常重要的，它利用用户的隐性反馈信息作为显性反馈信息的补充，建立矩阵分解模型。该模型的形式化描述如公式(3.7)所示：

$$\hat{r}(u,i) = \mu + b_u + b_i + q_i^T \cdot (p_u + |I(u)|^{-\frac{1}{2}} \cdot \sum_{j \in I(u)} y_j) \tag{3.7}$$

其中，μ 是观测数据的平均评分；b_u 是用户的评分偏置量，用来描述用户的评分偏好，是喜欢打高分还是喜欢打低分；b_i 是产品的评分偏置量，用来描述产品的平均得分相较于所有产品是偏高还是偏低；p_u 和 q_i 是矩阵分解后的用户向量和产品向量，*I(u)*是被用户 u 评过分的产品集合，y_j 是针对隐性反馈信息的产品因子。

对于以上参数，SVD++算法以最小化 Frobenius 损失函数为优化目标，使用随机梯度下降法进行训练。SVD++算法优化目标的形式化描述如公式(3.8)所示：

$$\begin{aligned} \min_{p*,q*,b*,y*} \sum_{<u,i>\in R} & (r_{ui} - \mu - b_u - b_i - q_i^T \cdot (p_u + |I(u)|^{-\frac{1}{2}} \cdot \sum_{j \in I(u)} y_j))^2 \\ & + \lambda_6 \cdot (b_u^2 + b_i^2) + \lambda_7 \cdot (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in I(u)} \|y_j\|^2) \end{aligned} \tag{3.8}$$

其中，λ_6 和 λ_7 是正则化因子，用来防止参数过拟合。基于该优化目标，SVD++算法利用观测数据矩阵 R 中的每个元素<*u,i*>进行参数训练；然后，将这些训练好的参数带入公式(3.7)预测用户对产品的评分，并将预测评分高的产品推荐给用户。

SVD++算法将缺失数据视为未知信息，在观测数据矩阵 R 上训练参数，学习模型。我们将利用 3.2 节提出的缺失数据建模方法改进 SVD++算法，提升其利用缺失数据的能力，进而改善推荐效果。本节的剩余部分将对这些改进方法进行详细介绍。

3.3.1 使用加权法改进的 SVD++算法

首先介绍的是使用加权法改进的 SVD++算法，该算法记做 WSVD++算法。WSVD++算法与 SVD++算法使用相同的评分预测模型（如公式(3.7)所示）。它们的主要区别在于数据的使用和损失函数的定义。

与 SVD++只使用观测数据训练模型不同，WSVD++同时使用观测数据和缺失数据。WSVD++按照公式(3.2)将默认评分 r_m 作为缺失数据的评分值，在统一数据模型 R^* 上进行训练。此外，WSVD++的损失函数是加权 Frobenius 函数，因此，其优化目

标应调整为公式(3.9)所描述的形式：

$$\begin{aligned}\min_{p*,q*,b*,y*}\sum_{<u,i>} w_{ui}\cdot((r_{ui}-\mu-b_u-b_i-q_i^T\cdot(p_u+|I(u)|^{-\frac{1}{2}}\cdot\sum_{j\in I(u)}y_j))^2\\ +\lambda_6\cdot(b_u^2+b_i^2)+\lambda_7\cdot(\|q_i\|^2+\|p_u\|^2+\sum_{j\in I(u)}\|y_j\|^2))\end{aligned} \tag{3.9}$$

WSVD++基于公式(3.1)定义的权重，利用 R^* 中所有数据训练推荐模型，以期最小化预测结果与真实结果的加权平方误差。对于给定数据<*u,i,r*>，我们定义预测误差为：

$$e_{ui}=r_{ui}^*-\hat{r}(u,i) \tag{3.10}$$

对于 R^* 中的每个数据，我们计算其预测误差，并依据此误差向梯度的反方向调整参数值，调整方法如下：

$$\begin{aligned}&\bullet b_u\leftarrow b_u+w_{ui}\cdot\gamma_1\cdot(e_{ui}-\lambda_6\cdot b_u)\\&\bullet b_i\leftarrow b_i+w_{ui}\cdot\gamma_1\cdot(e_{ui}-\lambda_6\cdot b_i)\\&\bullet q_i\leftarrow q_i+w_{ui}\cdot\gamma_2\cdot(e_{ui}\cdot(p_u+|I(u)|^{-\frac{1}{2}}\cdot\sum_{j\in I(u)}y_j)-\lambda_7\cdot q_i)\\&\bullet p_u\leftarrow p_u+w_{ui}\cdot\gamma_2\cdot(e_{ui}\cdot q_i-\lambda_7\cdot p_u)\\&\bullet\forall j\in I(u):y_j\leftarrow y_j+w_{ui}\cdot\gamma_2\cdot(e_{ui}\cdot|I(u)|^{-\frac{1}{2}}\cdot q_i-\lambda_7\cdot y_j)\end{aligned} \tag{3.11}$$

该方法可以使 WSVD++算法的参数向最小化加权 Frobenius 损失函数收敛。但是，这样调整参数值并不符合参数的设定初衷。b_u 和 b_i 分别是用户、产品的评分偏置量，代表着观测数据中用户或产品的评分倾向。而在统一数据模型中，r_m 被用于表示缺失数据中用户对产品的评分，但这些评分并不是用户显性给出的。因此，r_m 所代表的负例信息并不能反映用户或产品（被）评高分或低分的倾向，因而不应影响描述用户、产品评分偏置的参数，即 b_u 和 b_i。此外，y_i 是反映隐性反馈信息的参数，亦应不受缺失数据的影响。因此，我们为这三个参数设计了不同的权重函数，如公式(3.12)所示：

$$w'_{ui}=\begin{cases}1, & <u,i>\in R\\ 0, & <u,i>\notin R\end{cases} \tag{3.12}$$

相应的，WSVD++对参数的训练过程进行针对性调整，即观测数据会影响所有参数的训练，而缺失数据只影响用户向量 p_u 和产品向量 q_i 的训练，其影响能力受缺失数据的权重值 δ 的控制。训练过程的形式化描述如公式(3.13)所示：

$$
\begin{aligned}
&\bullet b_u \leftarrow b_u + w'_{ui} \cdot \gamma_1 \cdot (e_{ui} - \lambda_6 \cdot b_u) \\
&\bullet b_i \leftarrow b_i + w'_{ui} \cdot \gamma_1 \cdot (e_{ui} - \lambda_6 \cdot b_i) \\
&\bullet q_i \leftarrow q_i + w_{ui} \cdot \gamma_2 \cdot (e_{ui} \cdot (p_u + |I(u)|^{-\frac{1}{2}} \cdot \sum_{j \in I(u)} y_j) - \lambda_7 \cdot q_i) \\
&\bullet p_u \leftarrow p_u + w_{ui} \cdot \gamma_2 \cdot (e_{ui} \cdot q_i - \lambda_7 \cdot p_u) \\
&\bullet \forall j \in I(u) : y_j \leftarrow y_j + w'_{ui} \cdot \gamma_2 \cdot (e_{ui} \cdot |I(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_7 \cdot y_j)
\end{aligned}
\tag{3.13}
$$

WSVD++完成参数的训练后，使用公式(3.7)预测用户对产品的评分情况，并依据预测的评分值进行推荐。

3.3.2 使用抽样法改进的 SVD++算法

本小节介绍使用抽样法改进的 SVD++算法。依据抽样方法的不同，使用随机抽样法改进的方法记做 RSSVD++，使用近邻抽样法改进的方法记做 NSSVD++。

由于使用抽样方法从缺失数据中获取负例信息并不需要对负例信息进行加权处理，因此，RSSVD++、NSSVD++与 SVD++一样，以最小化 Frobenius 损失函数为优化目标（见公式(3.8)）。

与 WSVD++类似，在 RSSVD++和 NSSVD++中，参数 b_u、b_i 和 y_i 同样不应受到负例信息的影响。因此，我们设计指示函数，对正例信息和负例信息进行区分。正例信息会影响所有参数的训练，而负例信息只影响用户向量 p_u 和产品向量 q_i 的训练。指示函数的形式化定义如公式(3.14)所示：

$$
I_{ui} = \begin{cases} 1, & <u,i> \in R \\ 0, & <u,i> \in R^* - R \end{cases}
\tag{3.14}
$$

使用抽样法改进的 SVD++算法，利用统一数据模型 R^* 中的所有数据进行训练，以期最小化预测的平方误差。其参数训练过程为：

$$
\begin{aligned}
&\bullet b_u \leftarrow b_u + I_{ui} \cdot \gamma_1 \cdot (e_{ui} - \lambda_6 \cdot b_u) \\
&\bullet b_i \leftarrow b_i + I_{ui} \cdot \gamma_1 \cdot (e_{ui} - \lambda_6 \cdot b_i) \\
&\bullet q_i \leftarrow q_i + \gamma_2 \cdot (e_{ui} \cdot (p_u + |I(u)|^{-\frac{1}{2}} \cdot \sum_{j \in I(u)} y_j) - \lambda_7 \cdot q_i) \\
&\bullet p_u \leftarrow p_u + \gamma_2 \cdot (e_{ui} \cdot q_i - \lambda_7 \cdot p_u) \\
&\bullet \forall j \in I(u) : y_j \leftarrow y_j + I_{ui} \cdot \gamma_2 \cdot (e_{ui} \cdot |I(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_7 \cdot y_j)
\end{aligned}
\tag{3.15}
$$

RSSVD++算法和 NSSVD++都是使用抽样法改进的 SVD++算法，它们皆可使用

以上方法进行参数的学习和模型的训练。完成训练后，它们使用公式(3.7)预测用户评分，并依此向用户推荐产品。

3.4 实验与分析

3.4.1 实验设置

本章使用 MovieLens 的 100K 数据集设计验证实验。这个数据集包含了 943 个用户对 1682 部电影的 100000 条评分记录。整个数据集被随机分为 5 份，分别是 C1.test，C2.test，C3.test，C4.test 和 C5.test。每次实验使用其中 4 份作为训练集，推荐算法使用这部分数据生成推荐结果；另外 1 份作为测试集，验证推荐效果。对于测试集，我们进一步将其随机分为两等份，一份用于调整推荐算法的参数，另一份用于验证调整参数后的算法的推荐效果。实验共进行 5 次交叉验证。

为了评价算法有效性，我们使用 1.2.4 节中提到的 NDCG、1-Call 和 Recall 评价方法对算法的推荐准确率进行评价。其中，NDCG 和 1-Call 是比较常用的 Top-*N* 推荐准确率评价方法。Recall 是针对本章研究的目标问题——数据稀疏性问题的有效评价方法。有研究表明，在数据稀疏背景下，无论缺失数据是否是随机缺失的，Recall 评价方法都可以利用观测数据对数据全集做出无偏估计[167]。此外，我们使用 COV 和 CIL 评价方法对推荐结果的多样性和新颖性进行评价。NDCG+是针对推荐算法排序预测能力的评价方法，虽然不是 Top-*N* 推荐问题的核心指标，我们亦使用它作为对照评价方法，分析不同推荐算法的排序预测能力。

为了验证本章提出的缺失数据建模方法和相应推荐算法的效果，我们将 WSVD++、RSSVD++和 NSSVD++算法与 SVD++算法以及其它基线算法进行对比分析。本章实验共使用了六种基线算法，包括三种评分预测推荐算法（UserCF、Slope-one 和 SVD++），一种排序预测推荐算法（OrdRec），以及两种利用缺失数据的推荐算法（AllRank 和 Pure）。其中，UserCF 算法[5, 88]是一种基于近邻的协同过滤推荐算法，其基本思想是用户倾向于喜欢他的相似用户喜欢的产品。Slope-one 算法[94]是一种使用平均评分误差进行评分预测的推荐算法。SVD++算法[35]是推荐系统领域针对评分预测问题的最优算法之一。它亦是本章提出的改进算法的基础，通过对比该算法和本章提出的算法的推荐效果差异，可以明显地看出 3.2 节中介绍的缺失数据建模方法的有效性。OrdRec[124]是一种基于点序（Pointwise）的排序预测推荐算法。该算法亦是 SVD++算法的改进，利用该算法我们可以对比排序预测思想和缺失数据建模对推荐效果的改

进程度。PureSVD[115]和 AllRank-Regression[166]是两种利用缺失数据的推荐算法，它们是基于 SVD 模型的改进。为了降低原始推荐模型对推荐效果的影响，我们分别利用它们处理缺失数据的思路改进 SVD++算法，得到 Pure 和 AllRank 两种推荐算法。实验使用 Pure 和 AllRank 对比不同的缺失数据建模方法对推荐效果的影响。

本章实验将分别对比以上算法在 Top1、Top3 和 Top5 推荐中的 NDCG 和 Recall 的效果，以及它们在 Top5 推荐中的 1-Call、COV、CIL 和 NDCG+的效果。

为更有效的对比分析各推荐算法效果，我们通过实验为基线算法选择其表现最好的参数，参数选择过程在文中不再赘述，下面列出相应参数选择结果。UserCF 使用皮尔逊相似度作为衡量用户相似度的方法，并选择 50 作为近邻数。SVD++、Pure 和 AllRank 都选择 50 作为低维特征数，并进行 25 步迭代计算，正则化因子为 λ_6=λ_7=0.05，学习速率为 γ_1=γ_2=0.002。此外，AllRank 按照文献[166]中实验结果选择 r_m=2 和 w_m=0.05。OrdRec 对 50 维低维矩阵进行 60 步迭代，其正则化因子为 λ_6=0.005，λ_7=0.001，学习速率为 γ_1=γ_2=0.05，γ_3=γ_4=0.006。本章提出的 WSVD++、RSSVD++和 NSSVD++算法，与 SVD++使用相同的低维特征数、迭代次数、正则化因子和学习速率，以降低不同参数对推荐效果的影响。此外，本章提出的这三种改进算法加入了一些新的参数，下面首先将分析这些参数对推荐效果的影响；然后再对比分析本章提出的推荐算法和基线算法的推荐效果。

3.4.2 参数分析

首先，针对本章提出的推荐算法，通过实验分析不同参数设置对推荐效果的影响。在分析参数影响时，主要考虑推荐算法的推荐准确率。由于算法在不同推荐准确率评价方法上表现类似，我们只介绍各参数值对 NDCG 效果的影响。此外，对于一个算法包含多个参数的情况，当对一个参数进行分析时，需保持其它参数不变。

WSVD++是使用加权法对 SVD++算法的改进。该算法包含两个新增参数，δ 和 r_m。其中，δ 是缺失数据的权重值；r_m 是缺失数据的默认评分值。我们首先分析不同 δ 取值对 WSVD++算法推荐效果的影响。如 3.2 节所述，使用 r_m=0 作为默认评分值。

图 3.4 绘制了 C1.test 数据集上，WSVD++算法的 NDCG 表现随 δ 取值变化的曲线[12]。其中，δ 的取值从 0 增长到 1，以 0.1 为步长。

[12] WSVD++算法在其它几个数据集上的表现与此类似，同时，RSSVD++和 NSSVD++同样在各数据集上表现类似。因此，本小节以各算法在 C1.test 上的表现为例进行参数分析。

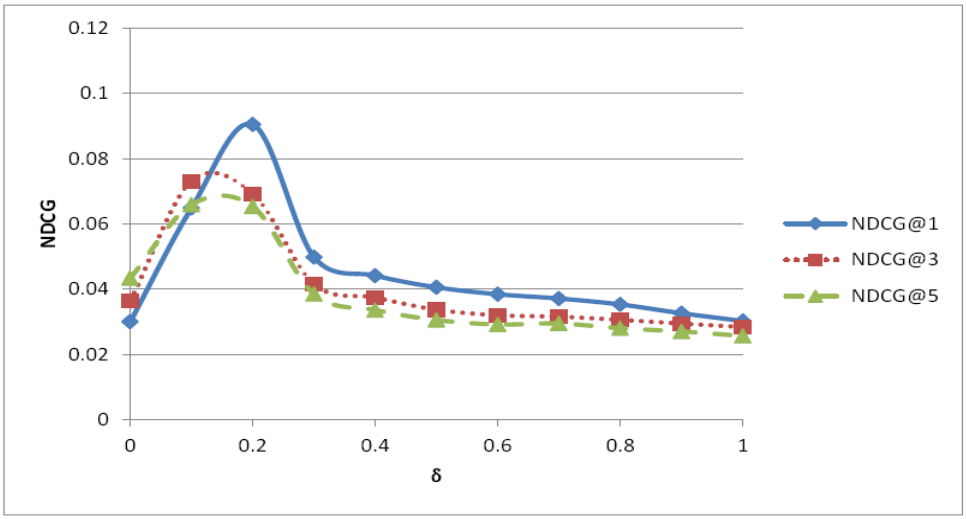

图 3.4 r_m=0 时 WSVD++的 NDCG 表现随 δ 变化的曲线

当 δ 取 0 时，相当于所有缺失数据并不会对推荐算法的训练起到任何影响，因此，WSVD++退化为 SVD++。当 δ 取值为 0.1 或 0.2 时，WSVD++的 NDCG 表现要远好于 SVD++。这说明使用加权法对缺失数据中的负例信息建模可以有效提升 SVD++算法的 Top-*N* 推荐准确率。但是，当 δ 取值大于等于 0.3 时，WSVD++的 NDCG 表现不再优于 SVD++。这就说明将缺失数据作为负例信息的置信度设置过大，并不利于提升 SVD++算法推荐相关产品的能力，也间接验证了全部缺失数据作为负例信息的低置信度。当 δ 取值为 1 时，WSVD++相当于是 Pure 算法。从图中可以看出，Pure 算法并不比 SVD++算法有更好的 NDCG 表现。综合来看，当 δ 取值为 0.2 时，WSVD++表现最好，因此，WSVD++选择 0.2 作为缺失数据权重值。

图 3.5 描绘了当 δ 为 0.2 时，WSVD++算法的 NDCG 表现随 r_m 取值变化的曲线。其中，r_m 的取值从-5 增长到 5，以 1 为步长。从图中可以发现，当 r_m 小于 1 时，WSVD++的 NDCG 表现较好。也就是说，当缺失数据的默认评分值在观测数据的评分范围之外时，WSVD++具有较好的向用户推荐相关产品的能力。该现象说明使用加权法处理缺失数据，并使用评分范围外的分值作为默认评分值是一种有效的缺失数据建模方法。此外，还可以发现，使用低评分作为缺失数据默认评分值对 SVD++算法的 NDCG 表现提升效果有限。这也说明了相对于将缺失数据当做低评分（Negative Ratings），将缺失数据当做负例信息（Negative Examples）是一种更有效的方法。综合来看，当 r_m 取值为 0 时，WSVD++获得了最佳的 NDCG 表现，因此，WSVD++选择 0 作为缺失数据的默认评分值。值得注意的是，当 r_m 小于 0 时，WSVD++的 NDCG 表现随 r_m 变

小而减少。这也许是由于缺失数据作为负例信息的置信度较低，过低的默认评分值会使推荐模型向负例方向偏移。

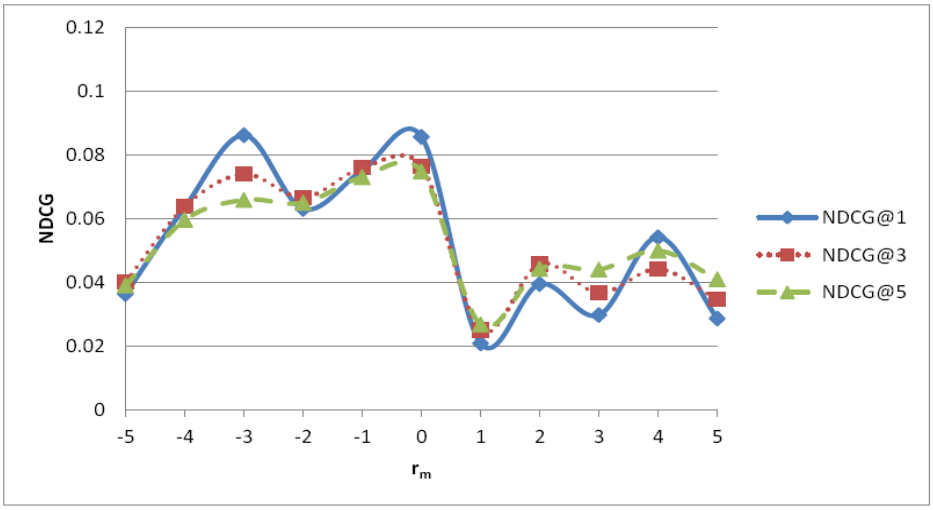

图 3.5 δ=0.2 时 WSVD++的 NDCG 表现随 r_m 变化的曲线

RSSVD++是使用随机抽样法对 SVD++算法的改进。该算法包含两个新增参数，θ 和 r_m。其中，θ 是从缺失数据中随机选取负例信息的比例；r_m 是缺失数据的默认评分值。首先，我们将分析不同 θ 对 RSSVD++算法推荐效果的影响。如前文所述，使用 r_m=0 作为默认评分值。

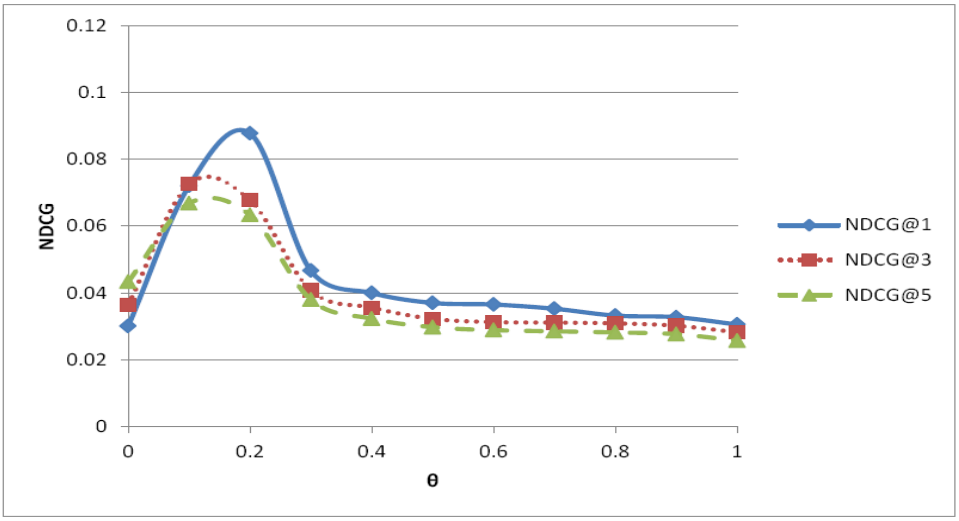

图 3.6 r_m=0 时 RSSVD++的 NDCG 表现随 θ 变化的曲线

图 3.6 绘制了 RSSVD++算法的 NDCG 表现随 θ 取值变化的曲线。θ 的取值从 0 增长到 1，以 0.1 为步长。

当 θ 为 0 时，RSSVD++无需从缺失数据中选取负例信息，RSSVD++退化为

SVD++。当 θ 取值为 0.1 或 0.2 时，RSSVD++的 NDCG 表现要远好于 SVD++。这说明使用随机抽样法从缺失数据中抽取负例信息，并将这些负例信息和观测数据一起用于训练推荐模型，可以有效提升 SVD++算法的 Top-N 推荐效果。但是，当 θ 取值大于等于 0.3 时，RSSVD++的 NDCG 表现不再优于 SVD++，说明从缺失数据中抽取过多数据作为负例信息，并不利于提升算法推荐相关产品的能力。这主要是因为，缺失数据中同时包含负例信息和未评分的正例信息，随机抽样使得抽取出的数据与原始缺失数据中正负例信息的比例相同，然而 RSSVD++将所有抽取出的数据都当做负例信息，这就使得用于训练推荐模型的数据中包含错误信息，而且错误信息的比例会随着抽样比例的增加而增加[13]。当错误信息对推荐模型的副作用大于新增负例信息对推荐模型的好处时，使用随机抽样法对 SVD++算法推荐效果的改善将逐渐下降。从图 3.6 中可以发现，当 θ 取值为 0.2 时，RSSVD++获得最好的推荐效果。因此，RSSVD++选择 0.2 作为从缺失数据中选取负例信息的比例。

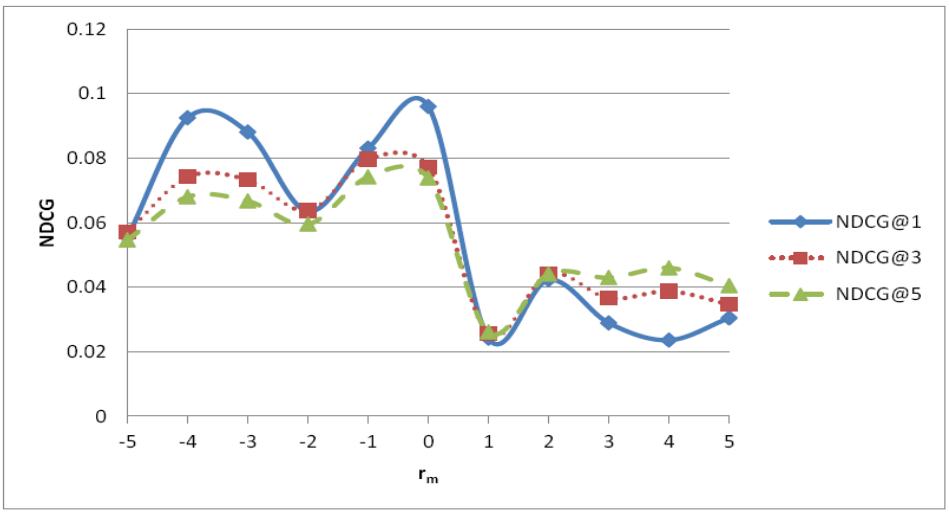

图 3.7 θ=0.2 时 RSSVD++的 NDCG 表现随 r_m 变化的曲线

图 3.7 绘制了当 θ 为 0.2 时，RSSVD++算法的 NDCG 表现随 r_m 取值变化的曲线。r_m 的取值从-5 增长到 5，以 1 为步长。从图中可以发现，当 r_m 小于 1 时，RSSVD++的 NDCG 表现较好。这说明使用随机抽样法从缺失数据中获取负例信息，并使用评分范围外的分值作为缺失数据的默认评分值是有效的，也再一次验证了相对于将缺失数据当作低评分信息，将其当作负例信息是一种更为有效的建模方式。综合来看，当 r_m 取值为 0 时，RSSVD++获得了最佳的 NDCG 表现，因此，RSSVD++选择 0 作为缺失

[13] 观测数据中的数据都为正确信息，抽样所得数据中包含固定比例的错误信息，因而随着抽样所得数据的增加，其占总训练数据的比例亦增加，这就导致了错误信息的比例随之增加。

数据的默认评分值。

NSSVD++是使用近邻抽样法对 SVD++算法的改进。该算法包含三个新增参数，K、θ 和 r_m。其中，K 是用户近邻数量，即用户近邻集合 $N(u)$的大小。随着 K 的增大，$N(u)$会随之增大，相应的，候选产品集合 $C(u)$会随之减小[14]。也就是说，候选产品集合的大小会随着 K 的取值而变化。θ 是从候选产品集合中随机选取负例信息的比例。因此，NSSVD++的负例样本集合是由 K 和 θ 共同决定的。此外，NSSVD++使用默认评分值 r_m 对负例信息进行填充。为了分析不同参数对 NSSVD++推荐效果的影响，我们使用三个典型的 K 值（20，50，80），使用 r_m=0 作为默认评分值，分析不同 θ 对 NSSVD++算法推荐效果的影响。

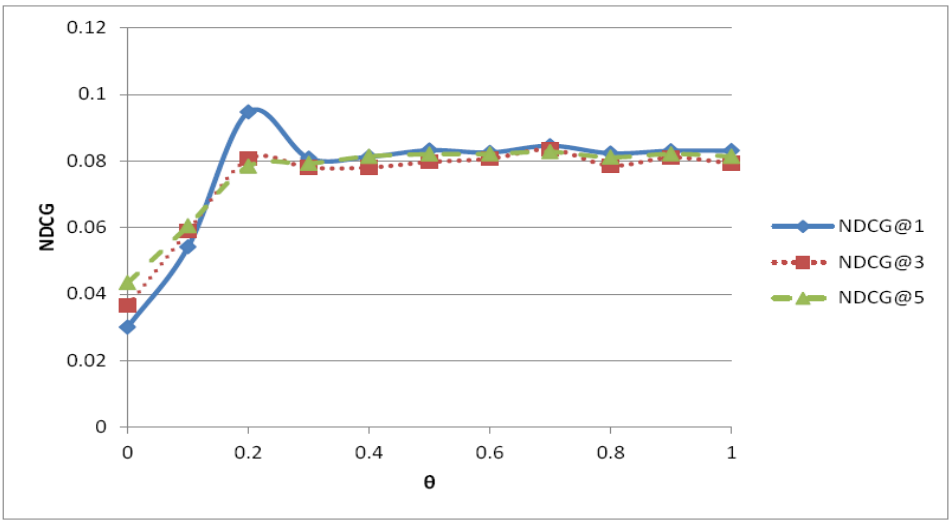

图 3.8 r_m=0，K=20 时 NSSVD++的 NDCG 表现随 θ 变化的曲线

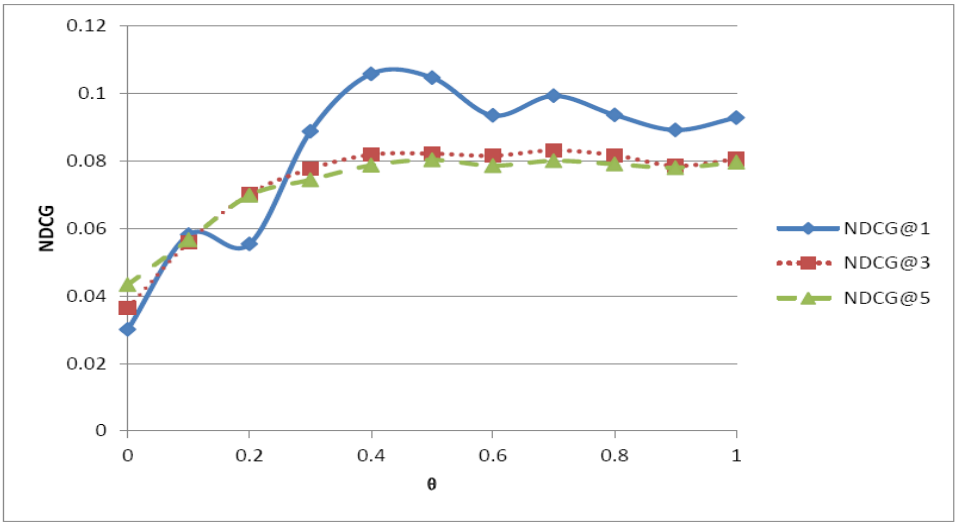

图 3.9 r_m=0，K=50 时 NSSVD++的 NDCG 表现随 θ 变化的曲线

[14] 越多的用户，未评分的产品会越少，相应的候选产品集合 $C(u)$会越小。

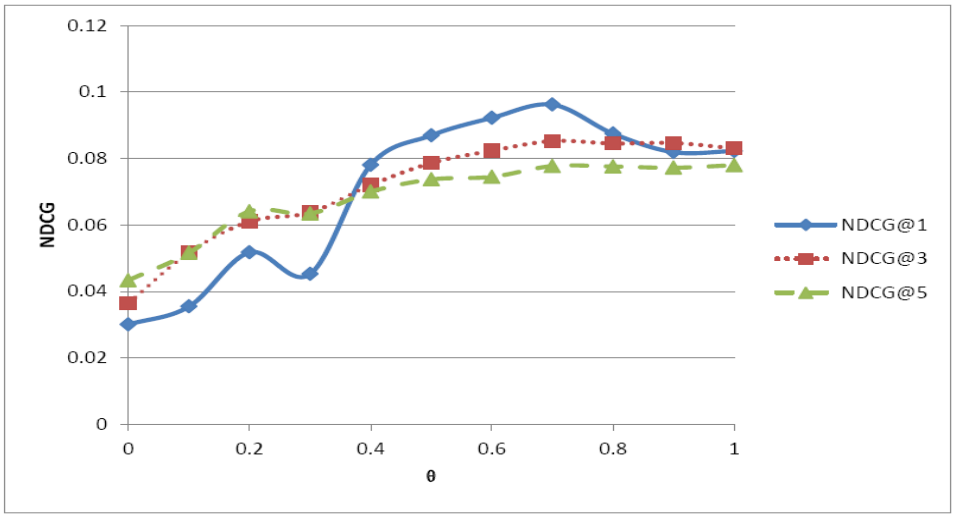

图 3.10 r_m=0，K=80 时 NSSVD++的 NDCG 表现随 θ 变化的曲线

图 3.8 到图 3.10 分别描绘了 K 取 20、50、80 时，NSSVD++算法的 NDCG 表现随 θ 取值变化的曲线。θ 的取值从 0 增长到 1，以 0.1 为步长。

当 θ 为 0 时，NSSVD++退化为 SVD++。当 θ 大于 0 时，无论 θ 的取值是多少，NSSVD++的 NDCG 表现都要好于 SVD++。这是一种完全不同于 WSVD++和 RSSVD++的现象。其主要原因是 NSSVD++使用了一种基于近邻的启发式抽样方法，利用相似用户的行为大幅提升了负例信息被抽样的可能性，降低了正例信息被抽样的可能性，保证了抽样所得的负例信息的可信度。对比图 3.8 到图 3.10，可以发现三幅图的一个共同特点，即当 θ 增长到一定值之后，NSSVD++的 NDCG 表现保持相对稳定，且稳定在一个较高水准。如当 K=20，$\theta \geqq 0.2$ 时，NSSVD++的 NDCG 表现保持在 0.08 左右；当 K=50，$\theta \geqq 0.4$ 时，NSSVD++的 NDCG 表现亦保持在 0.08 左右；当 K=80，$\theta \geqq 0.5$ 时，NSSVD++的 NDCG 表现保持稳定，但在 Top5 推荐中的 NDCG 表现略低于 0.08。对比三幅图中不同参数 NSSVD++的表现，当 K=50，θ=0.5 时，NSSVD++表现最好。这对参数即是 NSSVD++选定的近邻数和抽样比例值。

图 3.11 绘制了当 K 取 50，θ 为 0.5 时，NSSVD++算法的 NDCG 表现随 r_m 取值变化的曲线。r_m 的取值从-5 增长到 5，以 1 为步长。与 WSVD++和 RSSVD++相似，NSSVD++在 r_m 小于 1 时的 NDCG 表现远好于 r_m 大于等于 1 时。也就是说，将缺失数据当做负例信息是更有效的方式。与 WSVD++和 RSSVD++不同的是，当 r_m 取值在-5 到 0 之间时，NSSVD++的 NDCG 表现保持稳定。这也许是因为 NSSVD++使用基于近邻的启发式方式选取的负例信息具有较高的置信度。因此，r_m 的取值对推荐模型是否产生偏置影响不大。为了与 WSVD++和 RSSVD++保持一致，NSSVD++仍然选择 0

作为 r_m 的取值。

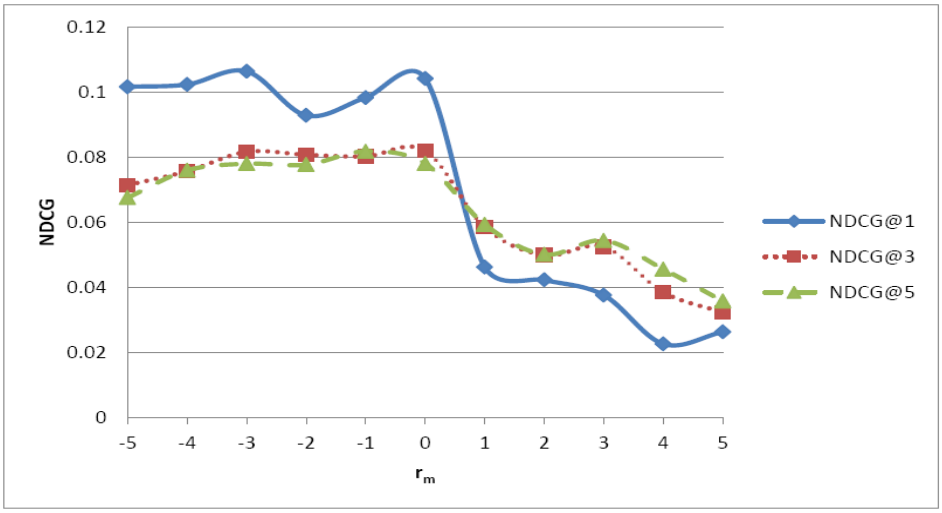

图 3.11 θ=0.5，K=50 时 NSSVD++的 NDCG 表现随 r_m 变化的曲线

3.4.3 实验结果

本小节将从推荐准确率、多样性和新颖性方面对比分析本章提出的算法和基线算法的推荐效果。表 3.1 中列出了这些算法推荐效果的对比结果。表中的每一行是一种算法，每一列是一种评价方法。对于每一种评价方法，分别加粗了该评价方法下表现最好的算法。

表 3.1 不同算法的推荐效果比较

Algorithms		NDCG			Recall			1-Call	COV	CIL	NDCG+
		1	3	5	1	3	5				
Benchmark	UserCF	0.019	0.019	0.020	0.000	0.002	0.003	0.12	**129.4**	**69.8**	0.65
	Slope-one	0.026	0.033	0.033	0.001	0.003	0.005	0.18	19.6	5.8	0.68
	SVD++	0.030	0.036	0.043	0.001	0.003	0.006	0.19	20.8	7.6	**0.71**
	OrdRec	0.080	0.065	0.062	0.002	0.005	0.008	0.23	19.0	4.6	0.52
	Pure	0.037	0.033	0.032	0.001	0.002	0.004	0.11	34.2	13.0	0.66
	AllRank	0.063	0.058	0.055	0.002	0.004	0.006	0.19	21.2	4.4	0.70
Proposal	WSVD++	0.104	**0.083**	**0.079**	0.004	0.008	0.014	0.28	23.6	8.8	0.70
	RSSVD++	0.095	0.074	0.073	**0.006**	0.007	0.013	0.27	24.0	9.2	0.69
	NSSVD++	**0.108**	**0.083**	0.078	**0.006**	**0.013**	**0.019**	**0.30**	38.2	16.2	0.68

表中第 3、4、5 列是不同算法在 Top1、Top3 和 Top5 推荐中的 NDCG 效果。其中，NSSVD++算法表现最好，其次是 WSVD++和 RSSVD++。计算可知，这些算法对 SVD++的改进至少为 68%。这说明本章提出的算法具有较好的 Top-N 推荐准确率。这

三种推荐算法都是基于缺失数据建模的改进型 SVD++算法，其良好表现验证了缺失数据中存在负例信息的观点，亦验证了相应缺失数据建模方法的有效性。此外，NSSVD++的 NDCG 表现要略强于 WSVD++和 RSSVD++，说明近邻抽样法启发式选择负例信息的方法是更为有效的缺失数据建模方法。

Pure 是基于将所有缺失数据都视为用户对产品评 0 分的思想对 SVD++进行改进的推荐算法。但是，其 NDCG 表现并未显著超越 SVD++算法。这说明简单的把所有缺失数据都视为高置信度的负例信息并不能有效改善算法的推荐准确率。AllRank 通过将缺失数据视为低评分数据的思想改进 SVD++算法。表 3.1 中的结果显示 AllRank 可以有效改善 SVD++算法的推荐准确率。但是对比 WSVD++、RSSVD++和 NSSVD++，AllRank 算法对 SVD++的改进程度较小。这说明将缺失数据视为负例信息是一种更有效的缺失数据建模方法。OrdRec 是一种排序预测的推荐算法，它通过点序排序的思想对 SVD++算法进行改进。近年来，排序预测被认为是比评分预测更贴近于 Top-*N* 推荐背景的推荐系统解决方案。表中结果亦显示，OrdRec 确实可以有效提升 SVD++算法的 Top-*N* 推荐准确率。但是，其提升效果并不如本章提出的算法效果显著。这表明利用缺失数据中的负例信息是一种比排序预测更有效的改善推荐算法 Top-*N* 推荐准确率的方式。

表中第 6、7、8 列是不同算法在 Top1、Top3 和 Top5 推荐中的 Recall 效果。Recall 是用来评价推荐算法向用户推荐相关产品能力的评价方法。该评价方法是 AllRank 算法的优化目标，但是，本章提出的三种推荐算法的 Recall 表现仍优于 AllRank 算法。这样的结果验证了利用缺失数据中的负例信息可以有效提升推荐算法向用户推荐相关产品的能力。

表中第 9 列是各算法在 Top5 推荐中的 1-Call 表现。与 NDCG 和 Recall 相似，同样是 NSSVD++算法表现最好，WSVD++和 RSSVD++表现次之。这三种算法的 1-Call 表现相较 SVD++算法提高了 42%以上，相较其它基线算法亦提高了 17%以上，说明本章提出的推荐算法有较好的保证 Top-*N* 推荐中至少有一个相关产品的能力。

表中第 10、11 列分别是不同算法的 COV 和 CIL 表现。其中，UserCF 表现最好。但是它的推荐准确率是所有算法中最差的。如果忽略 UserCF 算法，NSSVD++的 COV 和 CIL 表现要优于其它所有算法。同时，本章提出的三种算法都较 SVD++有所提升。以 SVD++算法为基准，这三种算法的 COV 表现至少提升了 13%，CIL 表现至少提升了 16%。此外，NSSVD++的表现要优于 WSVD++和 RSSVD++，再一次验证了近邻

抽样法的有效性。三种算法在 COV 和 CIL 评价方法上的良好表现，说明了它们在保证推荐准确率的前提下，还有效提升了推荐结果的多样性和新颖性。

表中第 12 列是不同算法的 NDCG+表现。在该评价方法中，SVD++算法表现最好，本章提出的算法表现较差。这主要是因为它们在推荐模型的训练中考虑了负例信息的影响，导致推荐算法在预测用户对产品评分情况时，既要考虑用户对产品的评分值信息，又要考虑用户是否会对产品评分。然而这种影响并不是非常显著，本章提出的算法相对于最佳算法 SVD++的表现也仅仅下降了 4%。

此外，观察实验结果可以发现，NDCG 表现好的算法 NDCG+表现未必好，反之亦然。这说明在观测数据中好的排序表现并不一定能带来好的 Top-*N* 推荐效果。因此，直接利用排序预测优化推荐模型并不一定可以有效提升推荐算法的 Top-*N* 推荐能力。

综上所述，本章提出的三种改进方法都有效改善了基线算法的推荐准确率和多样性，且其改进效果要优于其它对缺失数据建模的改进算法以及基于排序预测的改进算法。这样的实验结果验证了本章提出的缺失数据中包含用户兴趣的负例信息的观点，也验证了加权法、随机抽样法和近邻抽样法三种缺失数据建模方法和相应推荐算法的有效性。此外，三种改进算法中，NSSVD++效果最好，说明近邻抽样法启发式的抽取用户兴趣的负例信息是一种更为有效的方法。

3.5 本章小结

数据稀疏性是推荐系统面临的主要挑战之一。它主要表现在推荐系统包含的海量数据中往往只有少部分观测数据，大量数据是缺失的。为应对这个挑战，大多数研究假设缺失数据是随机地从数据全集中缺失的。因此，他们仅仅使用观测数据训练推荐模型，并利用训练好的模型预测用户行为，向用户推荐产品。然而，已有研究表明，缺失数据并不是随机缺失的。这使得大量研究的基础假设不再成立。

本章将部分缺失数据视作是由于用户主动选择不对产品进行评分而缺失的，并将这类缺失数据当作用户兴趣的负例信息，然后基于此观点，提出了三种对缺失数据建模并利用其中负例信息的方法，包括认为所有缺失数据都是低置信度负例信息的加权法；认为部分缺失数据是负例信息，通过随机抽样获取负例信息的随机抽样法；以及使用基于近邻的启发式抽样以获取负例信息的近邻抽样法。本章使用这三种缺失数据建模方法对 SVD++算法进行改进，分别提出了 WSVD++、RSSVD++和 NSSVD++推荐算法。实验结果表明本章提出的改进算法可以有效提升推荐算法在 Top-*N* 推荐中的

推荐准确率和多样性，验证了缺失数据并非随机缺失的观点，也验证了加权法、随机抽样法、近邻抽样法，以及相应推荐算法的有效性。

第4章 两步预测推荐算法

4.1 引言

推荐系统是帮助用户解决大数据时代信息过载问题的重要工具之一。用户对产品的评分信息是推荐系统的重要数据来源，亦是推荐系统为用户推荐产品的主要依据。在典型的基于用户评分的推荐系统中，系统收集用户对产品的评分行为信息，并依据用户对不同产品评分的高低，区分用户对产品的喜好程度。这些推荐系统往往认为用户对产品评高分代表用户喜欢这个产品，评低分代表用户不喜欢这个产品。因此，推荐系统依据用户的历史评分行为信息，识别用户的评分模式，以此预测其对未评分产品的评分值，并向用户推荐其可能评高分的产品。这种类型的推荐算法被称为评分预测推荐算法。

评分预测推荐算法一直是推荐系统研究领域关注的焦点，相关研究成果已广泛应用在一些实际系统中，为人们的学习、工作和生活提供着服务。评分预测推荐算法的研究者们一直致力于提升算法的评分预测准确率，认为这样可以提升用户对推荐结果的满意程度。但是，从用户角度看，他们并不需要推荐系统预测他们对产品的评分，而是希望推荐系统可以将他们感兴趣的产品呈现出来。因此，Top-*N* 推荐问题更符合推荐系统设计的真实目标。有研究表明，评分预测的准确程度和 Top-*N* 推荐效果之间并没有直接关系[115, 116]。所以简单的以提高评分预测准确率为目标，并不一定可以提升推荐系统的 Top-*N* 推荐效果。

在 Top-*N* 推荐中，推荐系统需要预测用户对产品的兴趣程度，并将用户最感兴趣的产品作为推荐结果呈现给用户，这可以看作是一个排序问题。因此，有研究者认为，相对于预测用户对产品的评分而言，预测用户对不同产品的排序可以更好的为用户进行 Top-*N* 产品推荐[117, 118]。近年来，学术界和产业界对排序预测推荐算法的研究逐渐升温。这种类型的推荐算法不再追求最小化评分预测的误差，而是以更好地预测用户对不同产品的排序为训练目标。有研究表明，排序预测推荐算法可以获得比评分预测推荐算法更好的 Top-*N* 推荐效果[116, 117]。

相对评分预测而言，排序预测确实更加贴近 Top-*N* 推荐问题的本质。但是，排序预测推荐算法是一种从表象解决 Top-*N* 推荐问题的方法。它们将用户对产品的评分信

息转换为点序（Pointwise）、对序（Pairwise）或者列表序（Listwise）的排序信息，然后选择并训练推荐模型，希望获得的推荐模型可以较好的拟合用户对产品的排序行为，排在靠前位置的产品可以很好地匹配用户兴趣。排序预测推荐算法一定程度上解决了 Top-*N* 推荐问题，但是通过这样表象地拟合用户对产品的排序模式，很难从语义层面对推荐结果进行解释，会降低用户对推荐结果的接受程度[92, 93]。

除此之外，无论是评分预测推荐算法，还是排序预测推荐算法，它们大多使用观测数据训练推荐模型，这些模型的有效性都是建立在用户是随机选择产品并进行评分的潜在假设上。而这一假设往往并不成立。第 3 章在介绍推荐系统中的数据稀疏性问题时，研究了推荐系统中缺失数据产生的原因，发现推荐系统中的缺失数据并不是随机缺失的，其中，部分缺失数据是由于用户主观选择不对相应产品评分而产生的。这说明用户随机选择产品并进行评分的假设并不成立。因此，简单的评分预测或者排序预测都不能很好的描述 Top-*N* 推荐问题。

有效描述用户兴趣，并依此建立用户兴趣模型是 Top-*N* 推荐需解决的关键问题之一。但究竟什么是用户感兴趣的产品呢？传统的推荐算法（无论评分预测推荐算法还是排序预测推荐算法）认为用户评高分的产品是用户喜欢的产品，用户对产品评低分则代表不喜欢这个产品。这一思想符合推荐系统设计评分体系的初衷，但是并不符合信息过载的时代背景。

在信息过载背景下，用户-产品评分矩阵极其稀疏，用户浏览并评分的产品只占产品集合的很小一部分，因此，用户在选择对哪些产品评分时，即为这一行为赋予了大量的偏好信息。本章从这个角度探索用户在推荐系统中的行为，提出一种两阶段用户行为模式，将用户对产品的评分行为分为两个阶段，即先依据兴趣选择待评分产品，然后对选定的产品给出评分值。这是一种全新的用户行为模式，它与传统推荐算法的最大区别是不再基于用户随机选择产品进行评分的假设，而是将用户选择待评分产品的过程重视起来。4.2 节将对两阶段用户行为模式进行详细介绍，分析其与传统用户行为模式的区别，并利用真实的推荐系统数据集通过数据分析的形式对其有效性进行验证。在两阶段用户行为模式的基础上，4.3 节提出一种解决 Top-*N* 推荐问题的两步预测推荐算法框架，该框架分别预测用户对产品评分的概率和具体的评分值，然后整合两步预测的结果向用户推荐产品。基于该框架，4.4 节和 4.5 节分别提出基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。最后，我们构建实验，验证两阶段用户行为模式和两步预测推荐算法的有效性。

4.2 两阶段用户行为模式研究

4.2.1 两阶段用户行为模式

推荐系统的主要思想是通过对用户行为信息的采集与分析，识别用户行为模式，然后利用群体智慧将可能匹配用户偏好的产品推荐给用户，以此辅助用户解决信息过载问题。其中，识别用户行为模式是非常重要的一环。

评分预测推荐算法是一种典型的推荐系统解决方案。它们认为用户对产品评高分代表用户喜欢这个产品，评低分代表用户不喜欢这个产品。因此，推荐系统利用用户的历史评分行为信息，识别用户的评分模式，预测用户对未评分产品的评分值，然后向用户推荐他自己可能评高分的产品。评分预测推荐算法的成立基于一个潜在的假设，即用户是随机选择产品进行评分的[166]。因此，其用户行为模式可总结如下：用户随机选择产品并对产品进行评分，然后依据其对产品的喜好程度，给予产品不同的评分值。评分预测推荐算法以对用户评分值预测的准确率为优化和评价的目标，包括RMSE、MAE 等。

评分预测推荐算法在推荐系统的发展过程中贡献巨大。但是，这类算法所解决的评分预测问题并不是推荐系统的核心问题。相对而言，Top-*N* 推荐问题更接近推荐系统的本质目标。有研究者认为，排序预测推荐算法可以比评分预测推荐算法更好地为用户进行 Top-*N* 产品的推荐[117, 118]。近年来，学术界和产业界对排序预测推荐算法的研究逐渐升温。这种类型的推荐算法认为推荐系统中的用户行为模式是：用户随机选择产品，依照个人喜好程度对选择的产品进行排序，然后按照排序结果依次给予评分。排序预测推荐算法通常使用各种点序、对序或列表序的排序效果评价方式及其变形作为优化目标，训练推荐模型。与评分预测推荐算法相似，排序预测推荐算法亦依托于用户是随机选择产品进行评分的假设。

评分预测推荐算法和排序预测推荐算法是当今协同过滤推荐技术的两大主要分支。它们大多依托于用户随机选择产品进行评分的假设，认为用户的行为模式只包含一个阶段，即用户对随机选定的产品进行评分。我们将这种行为模式称为传统用户行为模式。但是，如第 3 章所述，用户并不是随机选择产品进行评分的，传统用户行为模式的前提假设并不成立。为此，本章拟探索更符合真实情形的用户行为模式。

推荐系统已经广泛应用在电子商务、新闻、电影、音乐等各应用领域。我们以电子商务为例，分析用户在推荐系统中的行为模式。在电子商务网站（如淘宝、亚马逊

等），用户会根据自身需要，查询、浏览、并购买所需要的商品。用户在收到商品后，会依据客服质量、产品质量、物流服务等多方面情况，对产品进行评分。在这样的背景下，用户在推荐系统中评分行为的产生实际上分为两个步骤：

第一步是用户选择产品并决定对其评分的过程。在这一步中，用户往往会根据自身兴趣，自主选择欲评分的产品。在现今的信息过载时代，用户不可能浏览所有的产品，因此在选择欲评分产品时会倾向于选择他感兴趣的产品。例如在电子商务网站中，用户对商品的评分依托于之前的所有选择商品的环节（包括查询、浏览和购买等），并需要为获取这一评分机会付出一定的成本。因此，每一次评分都基于用户的主观选择，这种选择欲评分产品的行为可以看作是用户兴趣的体现。

第二步是用户对其选择的产品进行评分的过程。在这一步中，用户往往会根据产品的质量、自己的满意度等多方面进行综合考虑，然后给出一个表示用户喜好程度的评分值。用户对某个产品评低分，并不一定代表用户对这类产品不感兴趣，也许只是对当前产品质量的否定。同样以电子商务为例，用户也许会因为对物流、客服等方面的不满而对某个商品评低分，如果推荐系统依此认为用户不喜欢该类商品，则会产生对用户建模的偏差；即使用户是对商品本身不满，这种不满也不一定会扩展到类似的商品上，降低用户对这类产品的兴趣。例如用户对某空气净化器净化效果不满，并对其评了低分，并不应影响他对防霾口罩等相关产品的需求和兴趣。如果推荐系统能够向用户推荐这些产品，反而可以提升用户对推荐结果的满意程度。

因此，在推荐系统中，用户的评分行为包含两层含义，一是用户会关注其感兴趣的产品类型，在购买商品、观看电影等行为后，为相应的产品给出评分；二是用户会给其喜欢的产品较高的评分，感兴趣但不喜欢的产品较低的评分。本论文将这样的用户行为模式称为两阶段用户行为模式。

在两阶段用户行为模式中，用户对产品的评分行为被划分为选择产品进行评分和给定评分值两个阶段。而传统的用户行为模式只包含用户对产品评分一个阶段。对比两种用户行为模式的特点，可以发现它们的主要区别包括以下三个方面：

1) 对评分行为信息的重要性的认可程度不同。传统用户行为模式认为用户是随机选择产品并进行评分的，因此认为评分行为本身所含信息量极低；而在两阶段用户行为模式中，评分行为是用户兴趣的主要体现方式，也是推荐算法分析用户兴趣并进行推荐的主要依据。由此可见，不同用户行为模式对评分行为信息的重要性的认可程度差别很大。近年来，已经有部分研究者认识到了评分行为的重要性。文献[91]对比

了在数据稀疏环境下，基于用户的协同过滤算法使用不同相似度方法的表现效果。他们将使用评分值作为计算用户相似度的主要依据的方法称为 Correlation（包括皮尔逊相似度、余弦相似度等），将使用评分行为作为计算用户相似度的主要依据的方法称为 Relevance（包括 Jaccard 相似度、对数似然相似度等）。其实验结果表明，在数据稀疏环境下，基于用户的协同过滤推荐算法使用 Relevance 作为相似度度量方式比使用 Correlation 效果更好。该结论说明了评分行为信息的重要性。此外，文献[35]将评分行为信息视为隐性反馈信息的一种，用其作为显性反馈（评分值）的补充，共同构建推荐模型。实验结果表明，使用隐性反馈可以提高推荐模型的推荐效果，说明了评分行为信息对提升推荐效果的重要性。以上研究成果从不同的方面说明了评分行为信息的重要性，也间接说明了从对评分行为重要性的认可程度看，两阶段用户行为模式较传统用户行为模式更有效。

2) 对观测数据和缺失数据产生原因的观点不同。传统用户行为模式认为，用户是随机选择产品进行评分的，因此，观测数据和缺失数据都是随机产生的。而两阶段用户行为模式认为，用户依据个人兴趣选择产品并进行评分，因此，观测数据是用户主观选择产生的，相应的，缺失数据亦然。第 3 章已经对缺失数据的缺失方式进行了深入的研究。研究结果表明，部分缺失数据是由于用户主动选择不对产品进行评分而产生的。这类缺失数据是用户兴趣的负例信息。相应的，观测数据则是用户兴趣的正例信息。换言之，评分行为本身就是用户兴趣的体现，该说法亦是两阶段用户行为模式中对用户行为的第一阶段的描述。因此，第 3 章的研究结果可以从观测数据和缺失数据产生原因的角度佐证两阶段用户行为模式的有效性。

3) 对评分信息，尤其是低评分信息的使用方式不同。在传统用户行为模式中，用户对产品评高分被认为是用户喜欢这个产品，用户对产品评低分则被认为是用户不喜欢这个产品。因此，对产品评低分的信息往往被当作用户兴趣的负面信息，使得推荐算法降低推荐相关产品的概率。而在两阶段用户行为模式中，无论用户对产品评高分还是低分，评分本身就代表用户对这一类型产品感兴趣。评分信息，即使是低评分信息仍可作为第一阶段的正面信息，用来扩展用户兴趣，提升为用户推荐相关产品的概率。4.2.2 节将针对用户评高分产品和评低分产品的相关性进行分析。如果相关性较低，说明用户评高分产品和评低分产品分别代表其感兴趣和不感兴趣的产品集合，则传统用户行为模式应更有效；反之，则说明用户评高分产品和评低分产品属于同类产品集合，评低分产品亦可作为用户兴趣的有效标识，相应的，两阶段用户行为模式应

更有效。

接下来，4.2.2 节将使用真实的推荐系统数据集，对用户评高分产品和评低分产品的相关性进行数据分析，评价不同用户行为模式使用用户评分信息的方式，进而验证两阶段用户行为模式的有效性[15]。此外，本章将基于两阶段用户行为模式设计两步预测推荐算法框架及相应推荐算法。这些推荐算法优秀的推荐效果亦可以验证两阶段用户行为模式的有效性。

4.2.2 数据分析

本节使用 GroupLens 研究团队在 MovieLens 网站上收集的用户对电影评分的真实数据（MovieLens 1M 数据集）进行分析。该数据集包含 6040 个用户对 3900 个产品的，共 1000000 条评分记录。所有评分都是 1-5 之间的整数，整个数据集上的平均评分值为 3.58。我们使用平均分 T_H=3.58 作为阈值，将数据集中的评分信息划分为高评分集合和低评分集合，对用户在两个集合中的评分历史信息进行相关性分析。

在 MovieLens 数据集中，电影被分为 18 种类型，包括动作（Action）、冒险（Adventure）、动画（Animation）、儿童（Children's）、喜剧（Comedy）、犯罪（Crime）、纪录片（Documentary）、剧情（Drama）、幻想（Fantasy）、黑色电影（Film-Noir）、恐怖片（Horror）、音乐剧（Musical）、悬疑（Mystery）、爱情（Romance）、科幻（Sci-Fi）、惊悚（Thriller）、战争（War）和西部片（Western）。每部电影属于一种或多种类型。我们使用电影类型做产品相关性评价标准，即如果两部电影属于同一类型，就认为它们是相关的，反之，则是不相关的。

首先，随机选取一个用户，观察其评分产品的电影类型分布。表 4.1 中列出了该用户的评分分布情况。对于一个电影属于多个类型的情况，用户对该电影的评分信息按照电影类型的个数平均分配到各类型的分布统计中。例如，致命武器（Lethal Weapon）同时属于动作、喜剧、犯罪和剧情四种类型，那么用户对其评 4 分的信息则会同时使这 4 种电影类型在被评 4 分的统计值上各增加 0.25。

[15]对于用户行为模式本身，无法进行有效的数据分析验证。此外，针对两种类型用户行为模式的区别中的前两点，已经有研究从数据分析及实验论证的角度进行了佐证，本章不再赘述。

表 4.1 MovieLens 数据集中一个随机用户的评分分布情况

GENRES	***Rating Values***						
	1	**2**	**3**	**4**	**5**	**Total**	**Percentage**
Crime	0.00	0.00	0.33	0.50	1.37	2.20	0.01
Animation	0.00	0.00	0.00	0.20	0.00	0.20	0.00
Romance	0.00	0.00	0.00	1.83	0.50	2.33	0.01
Horror	**1.50**	**5.83**	**38.33**	**43.53**	**45.07**	**134.27**	**0.68**
Comedy	**0.00**	**0.50**	**1.00**	**4.67**	**6.07**	**12.23**	**0.06**
Mystery	0.00	0.00	0.00	0.67	0.25	0.92	0.00
Action	0.00	0.83	1.17	2.37	3.15	7.52	0.04
Adventure	0.00	0.00	0.00	0.53	1.37	1.90	0.01
War	0.00	0.00	0.00	0.50	0.00	0.50	0.00
Sci-Fi	**0.50**	**0.83**	**7.00**	**5.70**	**5.62**	**19.65**	**0.10**
Musical	0.00	0.00	0.00	0.00	0.58	0.58	0.00
Thriller	**0.00**	**0.00**	**4.67**	**4.17**	**3.78**	**12.62**	**0.06**
Drama	0.00	0.00	0.50	2.33	0.25	3.08	0.02

从表 4.1 中可以看出，在 18 种电影类型中，该用户只对 13 种类型的电影有过评分记录，说明他对另外 5 种电影完全没有兴趣。本章使用阈值 R_T[16]作为区分用户是否经常对某一类型电影评分的参数。如果用户对某类型电影评分次数占其总评分次数的比例大于 R_T，则认为用户经常对该类电影评分。可以发现该用户经常评分的电影只包括恐怖片、喜剧、科幻和惊悚 4 种类型（表中加粗表示）。其中，恐怖片又占了绝大多数（总数的 68%），可以看出该用户是恐怖片的忠实粉丝。

我们使用 T_H 将这些评分信息划分为低分集合（Low-ratings）和高分集合（High-ratings），分别统计该用户的评分信息在不同集合中对应的电影类型分布。图 4.1 展示了其分布情况，其中蓝色柱形图（阴影填充）是低分集合中的分布情况，红色柱形图（纯色填充）是高分集合中的分布情况，绿色柱形图（无填充）是整个数据集中电影类型的分布情况。从图中可以看出，用户评高分多的电影类型，评低分也较多；评高分少的类型，相应的评低分也较少。而且这些分布与数据集中电影类型的整体分布情况也极不相同。也就是说，无论高分集合还是低分集合中，该用户总是倾向于观看并评价几个固定类型的电影。此外，在用户评低分的所有电影中，恐怖片大约占 72%的比例，如果使用传统的用户行为模式分析的话，很可能会认为用户不喜欢恐怖片，但很明显的可以看出这与事实是不相符的。

[16]本章使用 R_T=5%作为默认值。

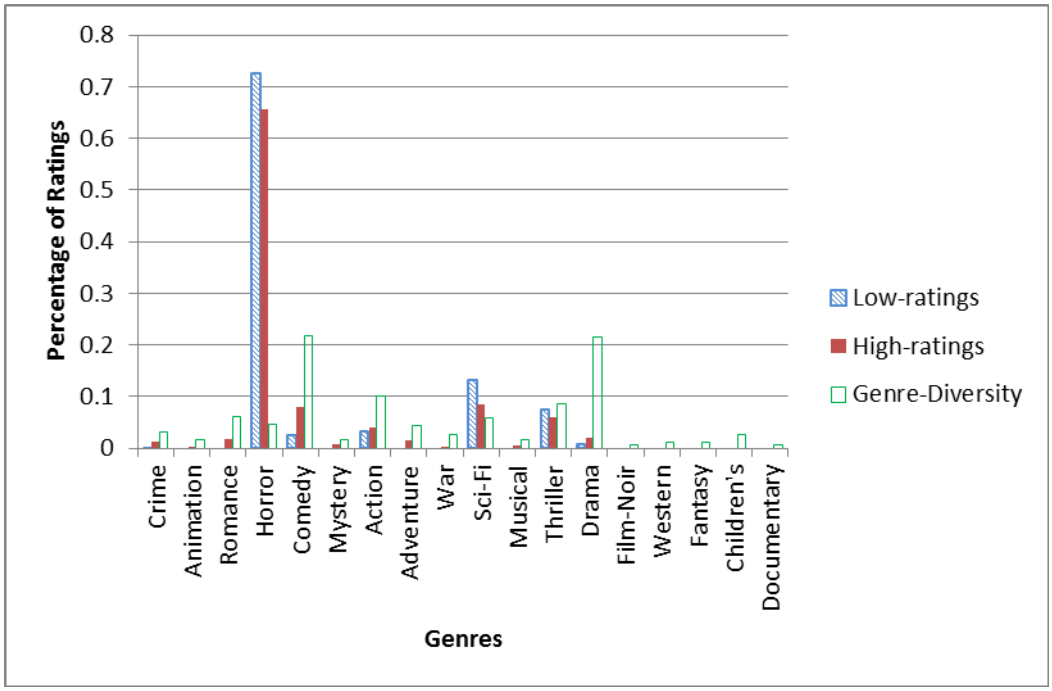

图 4.1 MovieLens 数据集中一个随机用户的评分分布统计情况

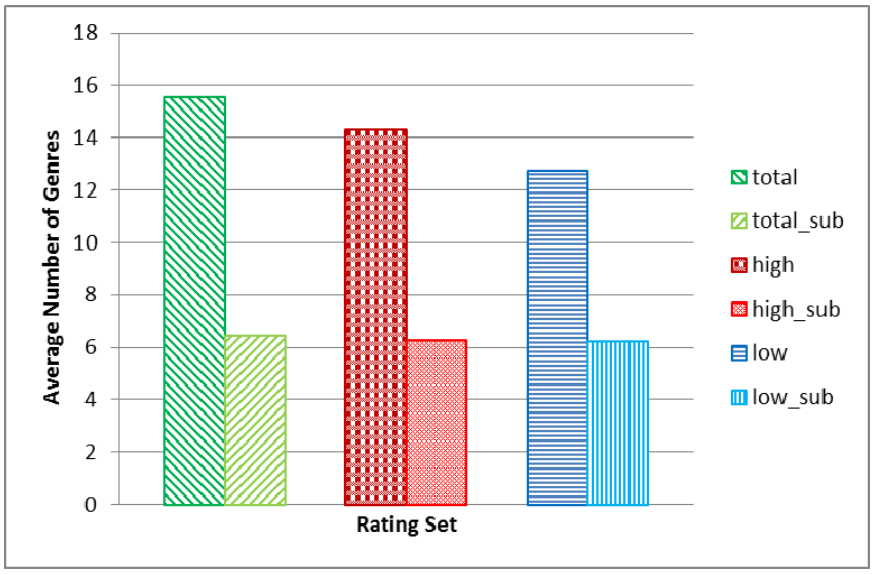

图 4.2 MovieLens 数据集中用户评分的电影类型的数目统计情况

其次，从数据集统计学特征上分析高分集合和低分集合中用户行为的相关性。图 4.2 展示了用户在不同集合上评分的电影类型数目的统计结果。其中，total、high 和 low 分别是整个数据集、高分集合和低分集合中平均每个用户评分的电影类型数。基于阈值 R_T，可以获取用户经常评分的电影类型的子集。我们统计所有用户经常评分的电影类型数，其结果用 total_sub 表示。相应的，按照数据集中高分集合和低分集合的分布

比例，设定阈值 R_H 和 R_L[17]，以获取用户经常评高分的电影类型子集和用户经常评低分的电影类型子集，并统计其平均数目，分别记为 high_sub 和 low_sub。

从图 4.2 中可以看出，用户平均只对 15.54 种类型电影有过评分记录，其中，经常评分的只有 6.47 种，它们分别只占 18 种电影类型的 86%和 36%。也就是说，用户经常评分的电影类型只占总数的大约三分之一，对其余的 14%从来不评分，50%很少评分。用户在 total、high 和 low 三个集合上评分的电影类型数分别为 15.54、14.30 和 12.72，total 是 high 和 low 的并集，那么 high 和 low 的交集的平均大小约为 11.47[18]。也就是说，平均每个用户会对 11.47 种类型的电影既评高分又评低分。该交集占 high 和 low 两个集合的比例分别为 80%和 90%，这说明 high 和 low 集合中的元素相似程度很高。如果我们只考虑用户经常评分的集合，即 total_sub、high_sub 和 low_sub，则如图 4.2 所示，集合相似程度更高。

最后，使用 Spearman 相关系数[19]针对每个用户的评分行为信息分析高分集合和低分集合的相关程度。对于任意用户，分别统计其在高分集合和低分集合中评分的电影类型分布情况，并在各自集合中将电影类型按照被评分次数做降序排列。对于电影类型 i，假设它分别处于高分集合和低分集合的第 H_i 和 L_i 位置，则其在两个集合中排序等级的差值为：

$$d_i = H_i - L_i \tag{4.1}$$

那么，该用户评高分集合和低分集合的 Spearman 相关系数为：

$$Spearman = 1 - \frac{6\sum_{i \in TC} d_i^2}{n(n^2 - 1)} \tag{4.2}$$

其中，TC 是该用户评过分的电影类型集合，n 是集合 TC 中元素的个数。

图 4.3 展示了整个数据集中所有用户在高分集合和低分集合中评分的电影类型的 Spearman 相关系数。图中用户按照 Spearman 相关系数降序排列。可以发现，94%的用户具有正的 Spearman 相关系数，77%的用户该系数值在 0.5 以上。这样的现象说明大多数用户在高分集合和低分集合中评分的产品具有很强的相关性。

[17] 在该数据集中，高分集合占 62%，低分集合占 38%。相应的，我们设定 R_H=0.62*R_T，R_L=0.38*R_T。

[18] 由容斥原理（Principle of Inclusion–Exclusion）可知，|high∩low|=|high|+|low|-|total|。

[19] Spearman 相关系数是分析两个序列相关性的经典方法之一，非常符合此处分析用户评高分集合和低分集合的电影类型的相关性的需求。详细介绍可以参见 http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient。

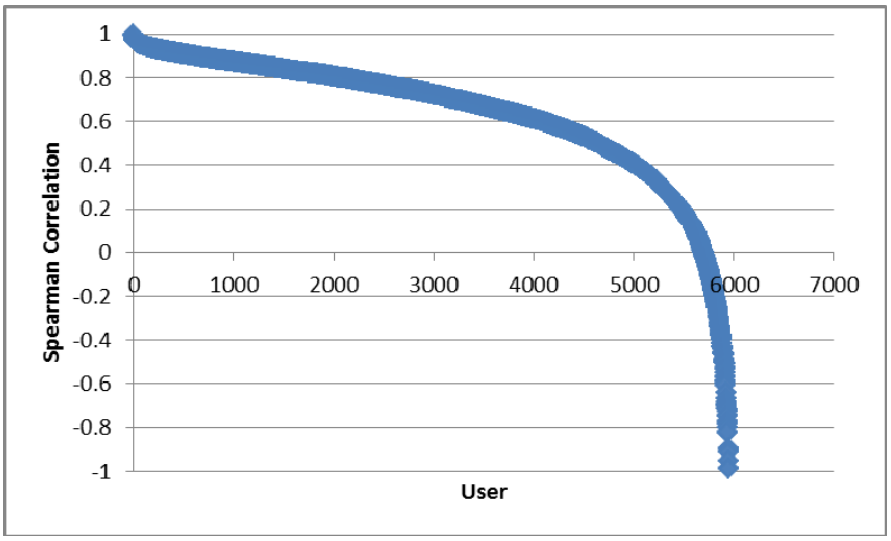

图 4.3 MovieLens 数据集中用户评分的电影类型的 Spearman 相关系数

综上所述，我们从样例用户、用户统计结果以及 Spearman 相关系数分布三个角度分析用户评分产品在高分集合和低分集合中的相关性。三个角度的结果都显示用户评高分产品和评低分产品具有较强的相关性。这些结果从对评分信息的使用方式上，验证了两阶段用户行为模式比传统用户行为模式更加有效。此外，文献[35]和[91]等研究验证了评分行为信息的重要性，第 3 章的研究结果说明了缺失数据并不是随机缺失的。以上研究使用数据分析、实验验证的形式，分别从两阶段用户行为模式和传统用户行为模式的三点主要区别说明了两阶段用户行为模式的有效性。

4.3 两步预测推荐算法框架

4.2 节介绍了两阶段用户行为模式，并通过数据分析验证了该用户行为模式的有效性。在两阶段用户行为模式中，用户对产品的评分行为被分为两个阶段。第一阶段，用户选择产品并决定对其评分；第二阶段，用户会按照喜好程度为其选择的产品给出一个合适的评分值。本节将在 4.2 节研究的基础上，探讨研究符合两阶段用户行为模式的推荐算法。

传统的推荐算法主要分为两种类型。一种是以预测用户评分模式为目标的评分预测推荐算法。该类型推荐算法认为用户对产品评高分则代表用户喜欢这个产品，评低分代表用户不喜欢这个产品。因此，推荐算法通过识别用户的评分模式，预测他对未评分产品的评分值，并向用户推荐其可能评高分的产品。另一种是以预测用户排序模式为目标的排序预测推荐算法。该类型推荐算法认为用户对两个或多个产品评分时，

会依据其喜好程度对产品进行排序，并按照排序结果依次给予评分。这两种类型的推荐算法都没有考虑用户是否会对产品进行评分，而是假设用户是随机选择产品进行评分的。因此，它们只针对用户对产品的评分值或者评分排序进行预测。这些假设和推荐解决方案并不符合两阶段用户行为模式的特点。

基于两阶段用户行为模式，我们设计并提出一种两步预测推荐算法框架。该框架通过对用户产生评分行为的过程进行仿真的方式预测用户的未来行为，并依此进行推荐。具体来说，第一步先预测用户对产品评分的概率，这是对两阶段用户行为模式第一阶段中用户选择欲评分产品过程的仿真；第二步预测用户对产品的评分值，这是对两阶段用户行为模式第二阶段中用户给出对产品评分值过程的仿真；然后推荐系统将结合两步预测的结果，把用户有较高概率评分并且会对其评高分的产品推荐给用户。两步预测推荐算法框架的整体框架结构如图 4.4 所示。

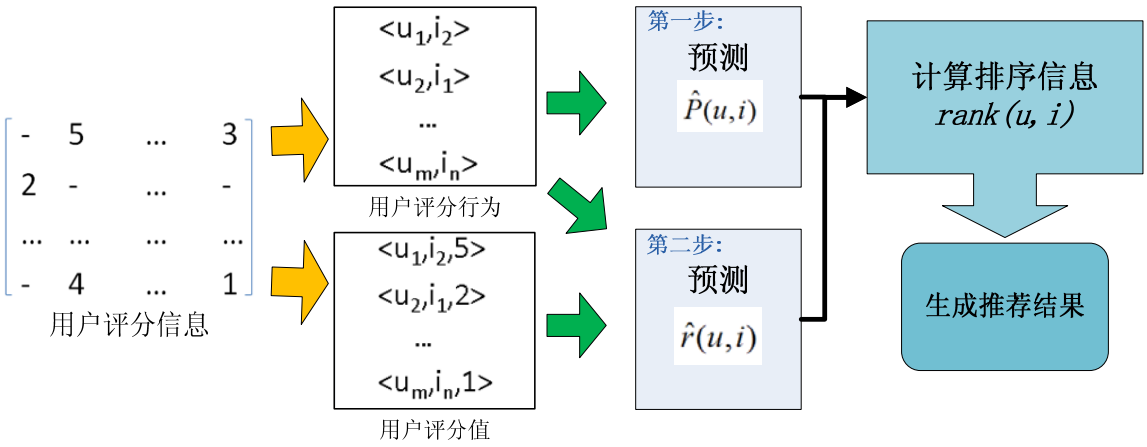

图 4.4 两步预测推荐算法框架

两步预测推荐算法框架依据两阶段用户行为模式将用户行为信息分为了用户评分行为（Rating behaviors）和用户评分值（Rating values）两个部分。其中，用户评分行为对应于两阶段用户行为模式中的第一阶段行为，即用户选择产品并进行评分的行为，这些行为是用户和产品的二元关系组，每个二元关系组<*u,i*>代表用户 *u* 已经对产品 *i* 有过评分记录；用户评分值对应于两阶段用户行为模式的第二阶段行为，即用户对其选择的产品给出的评分值，这些信息是用户、产品和评分值组成的三元关系组，每个三元关系组<*u,i,r*>代表用户 *u* 对产品 *i* 的评分值为 *r*。通常来看，用户评分值中包含了用户评分行为，因此被认为含有更丰富的信息，是大多数推荐算法偏爱使用的用户行为数据。然而，有研究表明，将用户评分行为从用户评分值中分离出来，可以有效提升推荐算法的效果[35, 91]。另外，在两阶段用户行为模式的第一阶段中，用户对

产品的具体评分值是无关信息，因此，将用户评分行为信息剥离开来，单独使用，也更符合两阶段用户行为模式的特点。4.6 节将通过实验验证将用户评分行为和评分值信息分割使用的有效性。

两步预测推荐算法框架的第一步是预测用户对产品评分的概率。这一步主要利用用户的历史评分行为信息，挖掘用户的兴趣偏好，预测用户可能会感兴趣的产品。用户对产品感兴趣的程度使用用户对产品评分的概率进行描述。由于仅仅预测用户对产品评分的概率，并不涉及相应的评分值信息，因此，需要的主要数据源为用户的历史评分行为信息，而无需用户的评分值信息。这一步的预测是本章所述推荐算法的核心部分，亦是它们和现有大多数研究的主要区别。如前文所述，大多数推荐算法认为用户是随机选择产品进行评分的，它们相当于默认用户对所有产品具有统一的评分概率。

第二步是预测用户对产品的评分值。这是推荐系统研究的经典问题之一，评分预测问题。该问题也是大量推荐算法研究的核心目标问题。可以充分利用现有研究成果解决第二步的评分值预测问题。该问题不是本章的主要目标问题，因此，本章中的推荐算法并未对评分预测算法提出新的改进。

本论文主要研究的是推荐系统中的 Top-N 推荐问题，该问题实质上是一个对候选产品集的排序问题。因此，在完成两步预测之后，两步预测推荐算法可以结合两步预测的结果，预测用户对产品的喜好的排序值，并依此排序结果向用户推荐产品。本章使用两步预测结果的乘积作为预测的排序值，其计算方法如下：

$$rank(u,i) = \hat{P}(u,i) \cdot \hat{r}(u,i) \tag{4.3}$$

其中，$\hat{P}(u,i)$ 是第一步预测的用户 u 对产品 i 评分的概率，$\hat{r}(u,i)$ 是第二步预测的用户 u 对产品 i 的评分值。此外，如果将用户不会对产品评分看作是用户对产品评 0 分，则用户对产品偏好的排序值计算方法可以写作：

$$\begin{aligned} & rank(u,i) \\ &= \hat{P}(u,i) \cdot \hat{r}(u,i) \\ &= \hat{P}(u,i) \cdot \hat{r}(u,i) + (1 - \hat{P}(u,i)) \cdot 0 \\ &= E[r(u,i)] \end{aligned} \tag{4.4}$$

因此，该排序值可以被视作用户对产品评分的数学期望。这可以作为两步预测推荐算法推荐产品的物理意义。

两步预测推荐算法框架是一个新颖的推荐算法框架。它将推荐问题分解为预测用

户对产品的评分概率，以及预测用户对产品的评分值两个步骤，是一个两步预测的推荐算法框架；它将用户评分数据分解为用户评分行为和用户评分值两个部分，分别对应两阶段用户行为模式的一个阶段，又是一个两阶段数据使用的推荐算法框架。此外，该推荐算法框架可以看作是用户产生评分数据的仿真过程，亦可以看作是对用户对产品评分的数据期望的预测。

文献[100]曾提出与本章类似的两步预测的观点。他们认为推荐系统中用户是自主选择产品并进行评分的。他们同样将推荐问题分解为预测用户对产品评分的概率和预测用户对产品的评分值两个部分，提出了基于自主预测的概率潜语义推荐算法（Probability Latent Semantic Analysis Model with Free Prediction，简称 PLSA_FP）。该推荐算法使用 *P(i|u)*定义用户 *u* 选择产品 *i* 进行评分的概率，使用 *P(r|i,u)*定义用户 *u* 对特定产品 *i* 评 *r* 分的概率，然后定义用户 *u* 选择产品 *i* 并评 *r* 分的概率为 *P(i|u)*和 *P(r|i,u)* 的联合概率，如公式(4.5)所示：

$$P(r, i \mid u) = P(r \mid i, u) \cdot P(i \mid u) \tag{4.5}$$

基于以上概率定义，PLSA_FP 使用 EM（Expectation Maximization）算法通过最大化观测数据产生概率训练推荐模型，进而向用户提供推荐服务。该推荐算法与本章提出的两步预测推荐算法都认可用户在推荐系统中产生的行为是分为两个步骤的，并且都以两步预测的形式向用户提供推荐结果。但是，两种算法之间存在着两点主要区别[20]，即：

1) PLSA_FP 是一种参数级的两步预测推荐算法，该算法使用参数将用户行为分为两个阶段，然后在一个统一的模型中以最大化观测数据产生概率为目标进行训练；而两步预测推荐算法框架是模型级的两步预测推荐算法，利用两种推荐模型分别预测用户对产品评分的概率以及用户对产品的评分值。

2) PLSA_FP 直接使用用户的评分值信息对推荐模型进行训练；而两步预测推荐算法框架将用户评分信息分解为评分行为信息和评分值信息，将其分别用于两步预测的不同阶段。

为了验证两步预测推荐算法框架的有效性，我们设计并提出了基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。这两个算法是两步预测推荐算法框架的具体实现，下面将对其进行详细介绍。

[20]本章的实验将使用 PLSA_FP 作为对比方法，分析这些区别对推荐效果的影响。

4.4 基于近邻的两步预测推荐算法

基于近邻的协同过滤推荐算法是协同过滤推荐技术的经典解决方案之一。它主要包括基于用户的协同过滤推荐算法和基于产品的协同过滤推荐算法两种类型。基于用户的协同过滤算法（UserCF）认为，一个用户会喜欢和他有相似爱好的用户喜欢的产品[5]。基于产品的协同过滤算法（ItemCF）认为一个用户会喜欢和他之前喜欢的产品类似的产品[6]。第 1 章已经详细介绍了这两种推荐算法，在此不再赘述。这两种推荐算法都使用基于用户评分的向量空间模型表示用户模型，并利用相似用户或者相似产品的评分信息预测用户对未评分产品的评分值。本节借鉴基于近邻的协同过滤算法的思想，将其与两步预测推荐算法框架相结合，提出基于近邻的两步预测推荐算法。下面将分别介绍基于用户的两步预测推荐算法和基于产品的两步预测推荐算法，它们分别是 UserCF 和 ItemCF 与两步预测推荐算法框架的结合。

4.4.1 基于用户的两步预测推荐算法

基于用户的两步预测推荐算法（User-based Two-step Collaborative Filtering，简称 UTCF）是经典的基于用户的协同过滤推荐算法和两步预测推荐算法框架的结合。

参照图 4.4 介绍的两步预测推荐算法框架，用户评分信息被分为用户评分行为和用户评分值。这两部分信息可分别用于构建相应的用户模型。其中，使用用户评分行为构建的用户模型称为二分用户模型（Binary User Model，简称 BUM）。该模型是一个 n 维的 0-1 向量（n 是推荐系统中的产品数量），1 代表用户已经对该产品评分，0 代表用户未对该产品评分。BUM 模型可以形式化的表示为：

$$V_{BUM}(u) = (v_1, v_2, \cdots, v_n) \tag{4.6}$$

其中，

$$v_i = \begin{cases} 1, & i \in I(u) \\ 0, & i \notin I(u) \end{cases} \quad (i \in [1, n]) \tag{4.7}$$

*I(u)*是用户 u 已评分的产品集合。使用用户评分值信息构建的用户模型即为传统的基于评分的用户模型（Rating User Model，简称 RUM），在此不再赘述。参照两步预测推荐算法框架，BUM 模型和 RUM 模型将被区分开来，应用到 UTCF 的不同阶段中。

(1) 第一阶段

在第一阶段中，UTCF 算法使用 BUM 模型预测用户对产品评分的概率。同 UserCF

推荐算法的思想类似，UTCF 算法认为用户会对与他相似的用户感兴趣的产品有兴趣。也就是说，用户倾向于评分的产品往往是他的相似用户已经有过评分行为的产品。因此，可以使用相似用户集合中对某一产品评分的人数预测用户对该产品感兴趣的程度，即：

$$Interest_{U_0}(u,i) = \sum_{a \in N(u)} V_{BUM}(a)[i] \tag{4.8}$$

其中，$V_{BUM}(a)[i]$是用户 a 的 BUM 模型中的第 i 个元素，如果取值为 1，则代表用户 u 对产品 i 有过评分；$N(u)$是用户 u 的相似用户集合。公式(4.8)利用相似用户的行为对当前用户的兴趣进行预测。在该方法中，相似用户集合中的所有用户对预测当前用户兴趣的影响力相同，并未受到用户相似度的影响。而实际上，相似度比较高的用户，其行为一致的可能性也比较大。因此，在预测用户评分行为时，可以使用用户相似度进行加权来有效描述这一特征。为此，我们将预测用户对产品感兴趣程度的方法调整如下：

$$Interest_U(u,i) = \sum_{a \in N(u)} sim(u,a) \cdot V_{BUM}(a)[i] \tag{4.9}$$

其中，$sim(u,a)$是用户 u 与用户 a 的相似度，公式(4.9)使用其取值作为预测用户 u 行为时衡量用户 a 行为影响力的权重。理论上，$sim(u,a)$可以使用任意相似度函数进行计算。但是，UTCF 算法第一阶段的目标是利用 BUM 模型预测用户对产品评分的概率，因此，使用用户评分行为计算用户相似度的 Relevance 方法比使用用户评分值计算用户相似度的 Correlation 方法更符合这一阶段的任务目标。为验证这个说法，我们将会在实验中对比使用 Relevance 方法和 Correlation 方法的 UTCF 算法推荐效果的区别。

UTCF 使用用户对产品的感兴趣程度预测用户对产品评分的概率，这就需要将 $Interest_U(u,i)$ 映射到[0,1]区间，一个直观的方法是使用 $Interest_U(u,i)$ 的最大值对其进行归一化，即：

$$\hat{P}_{UTCF}(u,i) = \frac{Interest_U(u,i)}{Max\_Interest_U(u)} \tag{4.10}$$

其中，$Max\_Interest_U(u)$ 是推荐算法对用户 u 的所有未评分产品预测的 $Interest_U(u,i)$ 的最大值，即：

$$Max\_Interest_U(u) = \max_{k \notin I(u)}(Interest_U(u,k)) \tag{4.11}$$

公式(4.10)的计算结果可以作为 UTCF 算法在第一阶段的预测结果。然而，UTCF 算法预测 $\hat{P}_{UTCF}(u,i)$ 的目的是计算 $rank(u,i)$，然后利用 $rank(u,i)$的排序结果为用户 u 选

择并推荐 N 个该用户可能会喜欢的产品。也就是说，保证 *rank(u,i)*针对用户 u 的排序结果是 UTCF 算法最关心的问题。结合公式(4.3)，可以发现，对于任意用户 u，$Max\_Interest_U(u)$ 是一个常量，在计算 $\hat{P}_{UTCF}(u,i)$ 时是否除以这样一个常量对 *rank(u,i)* 的排序结果并没有影响。所以，为了降低计算复杂度，可以在预测 $\hat{P}_{UTCF}(u,i)$ 时忽略这一常量。也就是说，UTCF 算法可以直接使用 $Interest_U(u,i)$ 对 $\hat{P}_{UTCF}(u,i)$ 进行预测，即：

$$\hat{P}_{UTCF}(u,i) = Interest_U(u,i) = \sum_{a \in N(u)} sim(u,a) \cdot V_{BUM}(a)[i] \tag{4.12}$$

(2) 第二阶段

第二阶段的目标是预测用户对产品的评分值。在 UTCF 中，我们分别使用两种策略进行预测。其中，一种是非个性化的预测方式，该方式利用产品的平均评分作为用户对未评分产品的预测评分值，即：

$$\hat{r}_{UTCF-M}(u,i) = mean(i) = \frac{\sum_{u \in U(i)} r_{ui}}{|U(i)|} \tag{4.13}$$

其中，*U(i)*是对产品 i 有过评分的所有用户的集合，r_{ui} 是用户 u 对产品 i 的评分值。

另一种策略利用基于用户的协同过滤算法预测用户对产品的评分值，该方法在第 1 章中已有了详细的介绍，在此不再赘述，其计算细节可参见公式(1.7)，该方法的预测结果记为 $\hat{r}_{UTCF-CF}(u,i)$。值得注意的是，该方法亦需要选择用户相似度计算方式，由于第二阶段是以预测用户评分值为目标的，因此，使用用户评分值计算用户相似度的 Correlation 方法更符合任务目标。同样，实验环节将对比在该阶段中使用 Correlation 相似度方法和 Relevance 方法的推荐效果。

UTCF 推荐算法结合两阶段的预测结果，使用公式(4.4)预测用户对产品评分的数学期望，并依此对候选产品进行排序，计算推荐结果。

4.4.2 基于产品的两步预测推荐算法

基于产品的两步预测推荐算法（Item-based Two-step Collaborative Filtering，简称 ITCF）是经典的基于产品的协同过滤推荐算法和两步预测推荐算法框架的结合。与 UTCF 类似，参照两步预测推荐算法框架，ITCF 在不同阶段分别使用 BUM 模型和 RUM 模型作为数据模型，完成相应的预测任务。

(1) 第一阶段

同 UTCF 相似，ITCF 推荐算法在第一阶段使用 BUM 模型预测用户对产品评分的

概率。借鉴 ItemCF 推荐算法的推荐思想（即，用户会喜欢与他之前喜欢的产品类似的产品），ITCF 认为用户倾向于评分的产品往往是与他之前评分的产品比较相似的产品。因此，在预测用户对某产品评分的概率时，可以使用目标产品与该用户已评分产品的平均相似度作为预测依据。也就是说：

$$\hat{P}_{ITCF}(u,i) \propto avg\_sim(u,i) = \frac{\sum_{j \in I(u)} sim(i,j)}{\left|I(u)\right|} \tag{4.14}$$

其中，*sim(i,j)*是产品 *i* 和产品 *j* 的相似度，*avg_sim(u,i)*表示产品 *i* 与用户 *u* 已评分产品的平均相似度。理论上，*sim(i,j)*可以使用任意计算产品相似度的方法进行计算。同 UTCF 类似，ITCF 推荐算法的第一阶段是利用 BUM 模型预测用户对产品评分的概率，因此，使用用户评分行为计算产品相似度的 Relevance 方法更符合这一阶段的任务目标。为验证这一说法，我们将会在实验中对使用 Relevance 方法和 Correlation 方法的 ITCF 算法的推荐效果进行对比分析。

由于推荐系统预测 $\hat{P}_{ITCF}(u,i)$ 的目的是为用户 u 选择并推荐 N 个该用户可能会喜欢的产品。因此，无需对 $\hat{P}_{ITCF}(u,i)$ 进行归一化。此外，|*I(u)*|对用户 *u* 来说，亦是一个常量，为降低计算复杂度，该值同样可以被忽略。因此，ITCF 推荐算法使用公式(4.15)预测用户 *u* 对产品 *i* 评分的概率：

$$\hat{P}_{ITCF}(u,i) = \sum_{j \in I(u)} sim(i,j) \tag{4.15}$$

(2) 第二阶段

与 UTCF 类似，ITCF 在第二阶段分别使用两种策略预测用户对产品的评分值。其中，非个性化的预测方式与 UTCF 相同，即使用公式(4.13)计算所得的产品平均评分进行预测，其预测结果记为 $\hat{r}_{ITCF-M}(u,i)$。另一种方式是利用基于产品的协同过滤推荐算法预测用户对产品的评分值。该方法在第 1 章中已有了详细的介绍，在此不再赘述，计算细节可参见公式(1.8)，该方法的预测结果记为 $\hat{r}_{ITCF-CF}(u,i)$。同样，该方法亦需要选择产品相似度计算方式，由于第二阶段是以预测用户评分值为目标的，因此，使用用户评分值计算产品相似度的 Correlation 方法更符合任务目标。同样，实验中将对比在该阶段中使用 Correlation 相似度方法和 Relevance 方法的推荐效果。

ITCF 推荐算法结合两阶段的预测结果，使用公式(4.4)预测用户对产品评分的数学期望，并依此对候选产品进行排序，计算推荐结果。

4.5 基于模型的两步预测推荐算法

经典的协同过滤推荐算法主要包括两个类型，一种是基于近邻的协同过滤推荐算法，该类推荐算法启发式地利用用户评分信息，寻找相似用户或相似产品，然后利用相似用户或产品的信息预测用户行为，并进行推荐；另一种是基于模型的协同过滤推荐算法，它们的基本思想是通过用户历史信息训练一个推荐模型，然后利用这个模型对用户行为进行预测和推荐。相对于基于近邻的协同过滤推荐算法，基于模型的协同过滤推荐算法往往具有更严谨的理论基础。这种类型的推荐算法是一种学习的方法，通过优化一个预先设定的目标，建立最优的模型，然后利用该模型进行推荐。

本节将介绍一种基于模型的两步预测推荐算法。该算法依据两步预测推荐算法框架，在各阶段分别选择适合完成相应预测目标的推荐模型，进行预测和推荐。两步预测推荐算法第一阶段的目标是预测用户对产品评分的概率，其对应的数据源是用户的评分行为信息，即 BUM 模型。由于 BUM 模型中只包含用户兴趣的正例信息，因此，生成模型是一种比较适合的推荐模型。另外，用户往往会对多种类型的产品有兴趣，产品也往往会属于多种类型，主题模型（Topic Model）可以满足表达这样丰富信息的要求。结合以上两点特征，我们在第一阶段中选择基于潜在狄里克莱分布（Latent Dirichlet Allocation，简称 LDA）的生成式主题模型作为预测用户对产品评分概率的模型。此外，LDA 模型还可以看作是一种概率图模型，其相应的概率语义也符合第一阶段预测用户对产品评分概率的目标。第二阶段是经典的评分预测问题，可以充分利用相关研究成果。我们在第二阶段中选择一种有效且应用广泛的矩阵分解模型 SVD++ 预测用户对产品的评分值。之后，整合两阶段的预测结果，计算用户对产品评分的数学期望，并进行排序和推荐。该算法是一种主题模型和矩阵分解模型的混合算法，本论文将其记做 HTMMF 算法（A Hybrid Approach of Topic Model and Matrix Factorization Based on Two-step Recommendation Framework）。下面分别对 HTMMF 算法的两个阶段进行详细介绍。

(1) 第一阶段

LDA 模型是文本处理领域非常有效的概率生成模型。它将文档（Document）看作是有限个不同主题（Topic）的概率混合，将主题看作是无限个词（Term）的概率混合。LDA 模型利用公式(4.16)描述文档的生成概率[98]：

$$P(w|d)=\sum_{z}P(z|d)\cdot P(w|z) \tag{4.16}$$

其中，*P(w|d)*是文档 *d* 已知情况下，词 *w* 出现的条件概率；*P(z|d)*是文档 *d* 属于主题 *z* 的条件概率；*P(w|z)*是主题 *z* 已知，词 *w* 出现的条件概率。LDA 模型将文档转换为词在不同主题上的联合概率分布，有利于对文本进行分类、自动摘要等相关处理。

LDA 模型是一种生成模型，可以只利用正例信息进行模型的学习和训练；它又是一种主题模型，可以表达一个用户有多种类型兴趣、一个产品属于多种类型的丰富信息；它还是一种概率图模型，可以表达用户和产品间的概率关系。它符合我们在第一阶段利用用户评分行为信息预测用户对产品评分的概率的各项要求。因此，我们在 HTMMF 推荐算法的第一阶段选择利用 LDA 模型预测用户对产品的评分概率。

类似于 LDA 模型在文本领域中加入主题维度，在推荐系统领域，LDA 模型在用户-产品二元关系中加入兴趣维度，将其变为用户-兴趣-产品的三元关系。相应的，用户对产品感兴趣的概率可以转换为用户在不同兴趣主题维度上对产品感兴趣的联合概率，即：

$$\hat{P}_{LDA}(u,i)=P(i|u)=\sum_{t}P(t|u)\cdot P(i|t) \tag{4.17}$$

其中，*P(t|u)*是用户 *u* 对兴趣主题 *t* 感兴趣的概率，*P(i|t)*是在兴趣主题 *t* 中选择产品时选中产品 *i* 的概率。

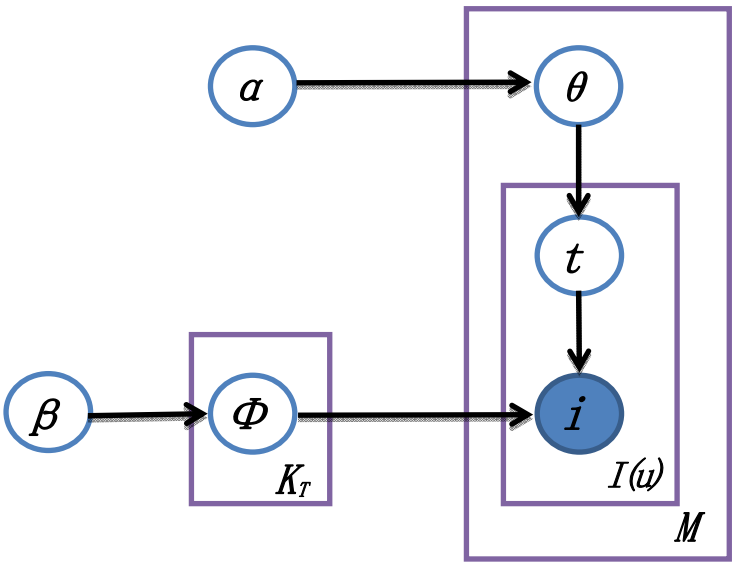

图 4.5 LDA 图模型表示

图 4.5 展示了推荐系统领域中 LDA 模型的概率图表示形式，其中实心圆代表观测变量，空心圆代表隐含变量。如图所示，在一个推荐系统中，共有 K_T 个兴趣主题，M 个用户，其中任意用户 u 包含 $I(u)$ 个已评分产品。i 是观测到的用户评分对象，t 是该产品的对应主题，θ 和 Φ 分别是用户兴趣主题和全局兴趣主题的概率分布参数，α 和 β 分别是 θ 和 Φ 的超参数（Hyper Parameter）。

在此概率图中，一个观测数据<u,i>的生成过程即为用户 u 选择产品 i 并决定评分的过程。其具体步骤主要包括两个物理过程。其中，一个是 $\alpha \rightarrow \theta \rightarrow t$ 的过程。在该过程中，对于用户 u，首先会依据 α 生成该用户的兴趣主题分布 θ_u，这一生成步骤服从狄里克莱分布；然后依据 θ_u 生成各观测变量的主题 t，这一生成步骤服从多项式分布。这一过程的物理意义是用户 u 依据个人兴趣分布选择兴趣主题 t。另一个是 $\beta \rightarrow \Phi \rightarrow i$ 的过程。在该过程中，首先依据 β 生成兴趣主题的概率分布 Φ，这一步骤服从狄里克莱分布；然后依据第一个物理过程中生成的观测变量的主题 t，选择 Φ 中的相应元素，生成观测变量 i，该步骤服从多项式分布。这一过程的物理意义是在选定的兴趣主题 t 中，用户 u 依据该主题的概率分布情况选择了产品 i。

以上过程是由已知隐含变量参数，生成观测数据的过程。该过程可以看作是 LDA 模型已知情况下进行预测和推荐的过程。因此，要进行预测和推荐，我们还需要完成 LDA 模型的训练工作。对于 LDA 模型的训练，有变分法（Varialtional Inference）[98] 和 Gibbs 抽样法（Gibbs Sampling）[168]等多种方式。本章主要介绍使用 Gibbs 抽样法的训练方式。

Gibbs 抽样法是马尔科夫链蒙特卡洛方法（Markov chain Monte Carlo）的一种形式，它通过最大化后验概率的形式训练 LDA 模型。具体而言，Gibbs 抽样法主要分为以下几个步骤：

1) 随机初始化：为所有用户评分行为中的产品随机分配一个初始的兴趣主题；
2) 重新扫描观测数据，对其中的每个评分行为记录，基于公式(4.18)利用观测数据中的其它行为信息，估计其条件概率，重新采样该产品的兴趣主题，更新相应模型参数；
3) 重复以上重新采样过程直到 Gibbs 抽样收敛；
4) 统计 Gibbs 抽样收敛后的兴趣主题-产品共现频率矩阵，利用公式(4.19)和(4.20)计算 LDA 模型的参数。

$$P(t_{ui} = t \mid \vec{t}_{\neg<u,i>}, \vec{i}) \propto \frac{CM_{ut,\neg<u,i>} + \alpha}{\sum_{k=1}^{K_T} CM_{uk,\neg<u,i>} + K_T \cdot \alpha} \cdot \frac{CN_{ti,\neg<u,i>} + \beta}{\sum_{j=1}^{n} CN_{tj,\neg<u,i>} + n \cdot \beta} \tag{4.18}$$

$$\theta_{ut} = P(t \mid u) = \frac{CM_{ut} + \alpha}{\sum_{k=1}^{K_T} CM_{uk} + K_T \cdot \alpha} \tag{4.19}$$

$$\Phi_{ti} = P(i \mid t) = \frac{CN_{ti} + \beta}{\sum_{j=1}^{n} CN_{tj} + n \cdot \beta} \tag{4.20}$$

其中，n 是推荐系统中产品的数量，t_{ui} 是用户 u 对产品 i 的评分记录对应的兴趣主题，CM_{ut} 是用户 u 的评分记录中的产品被抽样为兴趣主题 t 的个数，CN_{tj} 是产品 j 被抽样为兴趣主题 t 的次数，下标 $\neg<u,i>$ 是指忽略用户 u 对产品 i 的评分行为。

使用训练好的 LDA 模型参数，HTMMF 推荐算法利用公式(4.21)预测用户 u 对产品 i 评分的概率：

$$\hat{P}_{HTMMF}(u,i) = \sum_{t} P(t \mid u) \cdot P(i \mid t) = \sum_{t=1}^{K_T} \theta_{ut} \cdot \Phi_{ti} \tag{4.21}$$

实际上，已经有研究者将 LDA 模型应用到了推荐系统领域。但是他们直接使用用户评分值信息去预测用户对产品的兴趣程度[98]，这种方法记做 LDA_CF。文献[110]在 LDA_CF 的基础上，考虑了用户兴趣漂移的情况，该算法记做 iExpand。这两种算法都是直接使用用户评分值信息预测用户对产品的兴趣程度，并依此进行 Top-N 推荐。HTMMF 推荐算法与他们不同。HTMMF 推荐算法在两步预测推荐算法框架基础上，使用用户评分行为数据训练 LDA 模型，其目标是预测用户对产品评分的概率。然后在此预测结果的基础上结合评分预测信息，向用户推荐他们有较高概率评高分的产品。本章的实验使用 LDA_CF 和 iExpand 算法作为对比算法，验证两步预测推荐算法框架和 HTMMF 算法的有效性。

(2) 第二阶段

第二阶段的任务是预测用户对产品的评分值，这是经典的评分预测问题。HTMMF 推荐算法在第二阶段使用 SVD++模型预测用户对产品的评分值。

SVD++模型同时使用用户评分行为信息（BUM 模型）和用户评分值信息（RUM 模型）进行评分预测工作，该算法是一种有效且应用广泛的矩阵分解模型。3.3 节已

经对 SVD++模型有过介绍，它使用公式(3.8)进行模型训练，完成训练之后，使用公式(3.7)进行评分预测。具体内容可参看 3.3 节，在此不再赘述。

HTMMF 推荐算法结合 LDA 模型和 SVD++模型的预测结果，使用公式(4.4)计算用户对产品评分的数学期望，并进行排序和推荐。

4.6 实验与分析

4.6.1 实验设置

本章实验使用 MovieLens 的两个数据集进行验证。其中，一个数据集包含了 943 个用户对 1682 部电影的 100000 条评分记录，该数据集记为 ML1。另一个数据集包含了 6040 个用户对 3900 部电影的 1000000 条评分记录，该数据集记为 ML2。两个数据集中的评分值都是 1-5 之间的整数，5 表示最喜欢。

每个数据集都被随机分为 5 份。每次实验使用其中 4 份作为训练集，推荐算法使用这部分数据生成推荐结果；另外 1 份作为测试集，验证推荐效果。测试集进一步被随机分为两等份，一份用于调整推荐算法的参数，另一份用于验证调整参数后的算法的推荐效果。本节对每个数据集都进行 5 次交叉验证的实验。

为了评价算法有效性，实验使用 1.2.4 节中介绍的 NDCG、1-Call 和 Recall 评价方法对算法的推荐准确率进行评价。它们都是比较常用的 Top-*N* 推荐准确率的离线评价方法。其中，NDCG 综合考虑了 Top-*N* 推荐结果对测试集的覆盖程度、产品排序情况、以及用户对产品的评分信息，可以全面的描述算法的推荐准确率。因此，它被选为推荐准确率评价的主方法。同时，我们使用 COV 和 CIL 评价方法对算法的多样性和新颖性进行评价，并且选择 NDCG+进行对照。

本章的实验分为三个部分，第一部分（见第 4.6.2 节）对基于近邻的两步预测推荐算法进行实验分析，该部分主要对比 UTCF 和 ITCF 推荐算法使用不同类型的相似度函数和不同类型的评分预测算法的推荐效果；第二部分（见第 4.6.3 节）对基于模型的两步预测推荐算法进行实验分析，该部分主要分析不同参数值对 HTMMF 算法推荐效果的影响，并为 HTMMF 推荐算法选取适合的参数；第三部分（见第 4.6.4 节）将两步预测推荐算法与其它推荐算法进行比较，通过对比 UTCF、ITCF 以及 HTMMF 和基线算法的推荐效果，验证两阶段用户行为模式和两步预测推荐算法的有效性。

为了验证两步预测推荐算法框架，以及基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法的有效性，实验的第三部分共选择了七种基线算法进行推荐效果

的对比分析。其中，包括三种评分预测推荐算法（UserCF、ItemCF 和 SVD++），一种排序预测推荐算法（PLPA），两种使用 LDA 模型的推荐算法（LDA_CF 和 iExpand），以及一种参数级的两步预测推荐算法（PLSA_FP）。UserCF 和 ItemCF 是基于近邻的协同过滤推荐算法。UserCF[5]是基于用户的协同过滤推荐算法，其基本思想是用户倾向于喜欢他的相似用户喜欢的产品。ItemCF[6]是基于产品的协同过滤推荐算法，其基本思想是用户倾向于喜欢和他过去喜欢的产品相似的产品。SVD++算法[35]是推荐系统领域针对评分预测问题的最优算法之一，它也是 HTMMF 推荐算法第二阶段使用的推荐模型。PLPA[117]是一种基于概率主题模型，以对序（Pairwise）排序结果为优化目标的排序预测推荐算法。LDA_CF[98]和 iExpand[110]是两种使用 LDA 模型的推荐算法。其中，LDA_CF 使用用户评分值信息预测用户对产品的兴趣程度。iExpand 在 LDA_CF 的基础上，考虑了用户兴趣漂移的情况。PLSA_FP[100]是一种参数级的两步预测的推荐算法，4.3 节中已经详细描述了它与两步预测推荐算法框架的区别，在此不再赘述。

对于以上算法我们分别计算它们在 Top1、Top3 和 Top5 推荐中的 NDCG 和 Recall 的效果，以及它们在 Top5 推荐中的 1-Call、COV、CIL 和 NDCG+的效果。

为更有效的对比分析各推荐算法效果，我们通过实验为基线算法选择其表现最好的参数，参数选择过程在文中不再赘述，下面列出相应参数选择结果。UserCF 和 ItemCF 使用皮尔逊相似度作为衡量相似度的方法，并选择 50 作为近邻数。UTCF 和 ITCF 亦使用 50 作为近邻数量。SVD++选择 50 作为低维特征数，并进行 25 步迭代计算，正则化因子为 λ_6=λ_7=0.05，学习速率为 γ_1=γ_2=0.01。iExpand 使用文献[110]中为 MovieLens 数据集优化的参数结果，即 300 个潜在兴趣主题并进行 1000 步迭代，参数 α、β、s 和 c 的取值分别是 0.001、0.08、1 和 0.9。LDA_CF 与 iExpand 使用相同的 LDA 模型参数，包括 α、β、K_T和迭代次数。PLPA 使用文献[117]建议的参数取值，即 6 个潜在兴趣主题和 30 步迭代。PLSA_FP 的参数使用文献[100]建议的 40 个潜在兴趣主题和 30 步迭代步数。

4.6.2 基于近邻的两步预测推荐算法实验分析

本小节针对基于近邻的两步预测推荐算法，构建实验分析相似度函数和评分预测策略对推荐效果的影响。在分析这两方面影响时，主要考虑推荐算法的推荐准确率。在推荐准确率评价方法中，选用主评价方法 NDCG 进行评价。

相似度函数主要包括两种类型，一种以用户评分值作为计算相似度的主要依据，该方法称为 Correlation；另一种以用户评分行为作为计算用户相似度的主要依据，这

种方法称为 Relevance。我们分别选择皮尔逊相似度（见公式(1.2)）和对数似然相似度（见公式(1.4)）作为 Correlation 方法和 Relevance 方法的样例，分析不同类型相似度函数对基于近邻的两步预测推荐算法的推荐效果的影响。其中，使用 Relevance 相似度函数相当于在第一阶段仅使用了用户评分行为的信息，是符合两步预测推荐算法框架的方法；而使用 Correlation 相似度函数则相当于使用了用户的评分值信息，该方法用于对比验证两步预测推荐算法框架在第一阶段中只使用用户评分行为信息的要求。

基于近邻的两步预测推荐算法提出了两种评分预测的策略，一种是使用非个性化的产品平均分进行预测的方式，另一种使用传统的基于近邻的协同过滤推荐算法进行预测。我们将对比分析这两种策略对推荐效果的影响。此外，还需要对比基于近邻的协同过滤推荐算法使用不同类型相似度函数的效果。因此，在第二阶段的评分预测问题中，共需分析三种不同的情形：即使用非个性化的产品平均分进行预测的方式，记为 M；使用 Relevance 相似度函数进行评分预测的方式，记为 R；以及使用 Correlation 相似度函数进行评分预测的方式，记为 C。UTCF 推荐算法和 ITCF 推荐算法在使用基于近邻的策略进行评分预测时分别使用相应的 UserCF 或 ItemCF 推荐算法。

针对以上两方面的影响因素，将其两两组合，分别记为 Rel-M，Cor-M，Rel-R，Cor-R，Rel-C 和 Cor-C。其中，Rel 和 Cor 分别代表在第一阶段使用 Relevance 相似度函数（即对数似然相似度）和 Correlation 相似度函数（即皮尔逊相似度）。下面将分别对 UTCF 和 ITCF 进行实验分析。

(1) UTCF 推荐算法的实验分析

首先分析不同相似度函数和评分预测方法对 UTCF 算法推荐效果的影响。表 4.2 和表 4.3 分别展示了 ML1 数据集和 ML2 数据集上，UTCF 推荐算法的 NDCG 表现。

表 4.2 ML1 数据集上 UTCF 推荐算法的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
UTCF	Rel-M	**0.2704**	**0.2352**	**0.2185**
	Cor-M	0.1098	0.0896	0.0837
	Rel-R	0.2612	0.2219	0.2093
	Cor-R	0.1023	0.0883	0.0829
	Rel-C	0.2675	0.2257	0.2109
	Cor-C	0.1046	0.0905	0.0832

表 4.3 ML2 数据集上 UTCF 推荐算法的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
UTCF	Rel-M	**0.3192**	**0.2846**	**0.2632**
	Cor-M	0.1077	0.0890	0.0814
	Rel-R	0.2981	0.2765	0.2499
	Cor-R	0.1009	0.0812	0.0675
	Rel-C	0.3003	0.2795	0.2587
	Cor-C	0.1034	0.0859	0.0707

从表中可以看出，无论是在 ML1 数据集中还是在 ML2 数据集中，UTCF 推荐算法在第一阶段使用 Relevance 相似度函数总是比使用 Correlation 相似度函数具有更好的推荐效果。这就说明两步预测推荐算法框架提出的在第一阶段预测中只使用用户评分行为信息是有效的。

此外，第二阶段中使用产品平均分和使用基于近邻的协同过滤算法的推荐效果相差不大，使用 Relevance 相似度和 Correlation 相似度相差亦不大。这说明了决定 UTCF 算法推荐效果的重点在于第一阶段的预测准确性。也就是说，使用非个性化的产品平均分作为评分预测结果可以获得和使用基于近邻的协同过滤推荐算法类似的推荐效果。由于基于近邻的协同过滤推荐算法相较产品平均分的计算复杂度更高，因此，Rel-M 方法具有较好的综合表现。

在 4.6.4 节的对比实验中，UTCF 推荐算法选择使用对数似然相似度作为相似度函数，在第二阶段使用产品平均评分作为评分预测方法。

(2) ITCF 推荐算法的实验分析

下面将分析不同相似度函数和评分预测方法对 ITCF 算法推荐效果的影响。表 4.4 和表 4.5 分别展示了 ML1 数据集和 ML2 数据集上，ITCF 推荐算法的 NDCG 表现。

与 UTCF 实验结果类似，无论是在 ML1 数据集中还是在 ML2 数据集中，ITCF 推荐算法在第一阶段使用 Relevance 相似度函数总是比使用 Correlation 相似度函数具有更好的推荐效果。这样的结果再一次验证了在两步预测的第一阶段预测中只使用用户评分行为信息的有效性。

ITCF 推荐算法在第二阶段中使用产品平均分和使用基于近邻的协同过滤算法的推荐效果相差不大，使用 Relevance 相似度和 Correlation 相似度相差亦不大。这说明

了决定 ITCF 算法推荐效果的重点在于第一阶段的预测准确性。该结果与 UTCF 的实验结果相似。因此，综合考虑推荐准确率和算法的计算复杂度，Rel-M 方法具有较好的综合表现。在 4.6.4 节的对比实验中，ITCF 推荐算法选择使用对数似然相似度作为相似度函数，在第二阶段使用产品平均评分作为评分预测方法。

表 4.4 ML1 数据集上 ITCF 推荐算法的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
ITCF	Rel-M	**0.0481**	**0.0469**	**0.0463**
	Cor-M	0.0008	0.0005	0.0004
	Rel-R	0.0386	0.0379	0.0375
	Cor-R	0.0000	0.0002	0.0001
	Rel-C	0.0400	0.0425	0.0405
	Cor-C	0.0000	0.0002	0.0001

表 4.5 ML2 数据集上 ITCF 推荐算法的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
ITCF	Rel-M	**0.1240**	**0.1201**	**0.1148**
	Cor-M	0.0002	0.0001	0.0001
	Rel-R	0.1092	0.1003	0.0986
	Cor-R	0.0002	0.0001	0.0001
	Rel-C	0.1176	0.1146	0.1015
	Cor-C	0.0004	0.0002	0.0002

4.6.3 基于模型的两步预测推荐算法实验分析

本小节主要针对基于模型的两步预测推荐算法 HTMMF 进行实验分析。该算法在第二阶段进行评分预测时使用的 SVD++算法是一种成熟的解决方案，可直接利用现有研究成果，无需对其细节进行探讨。因此，本小节主要分析 HTMMF 推荐算法在第一阶段使用 LDA 模型预测用户对产品评分概率时，不同的参数设置和不同的数据模型对推荐效果的影响。在分析影响时，选择最重要的评价指标——推荐准确率进行分析。由于推荐算法在不同推荐准确率评价方法上表现类似，因此，只介绍它在 NDCG 评价方法上的表现。

(1) 第一阶段参数设置的实验分析

首先通过实验分析不同参数设置对推荐效果的影响[21]。HTMMF 算法在第一阶段使用 LDA 模型，该模型共包含超参数 α 和 β、兴趣主题个数 K_T、以及迭代步数 *iter* 四个参数。文献[169]对 LDA 模型的参数设置进行了细致研究，其研究结果表明，LDA 模型在 $\alpha = 50/K_T$、β=0.01 时效果较好。我们在此研究的基础上，研究不同参数设置对 HTMMF 算法推荐效果的影响。由于 α 的取值由 K_T 决定，因此，只需分析 K_T 和 *iter* 的不同取值对 HTMMF 算法推荐效果的影响。在分析 K_T 对推荐效果的影响时，需保持 *iter* 取值不变；反之，分析 *iter* 对推荐效果影响时需要 K_T 保持不变。

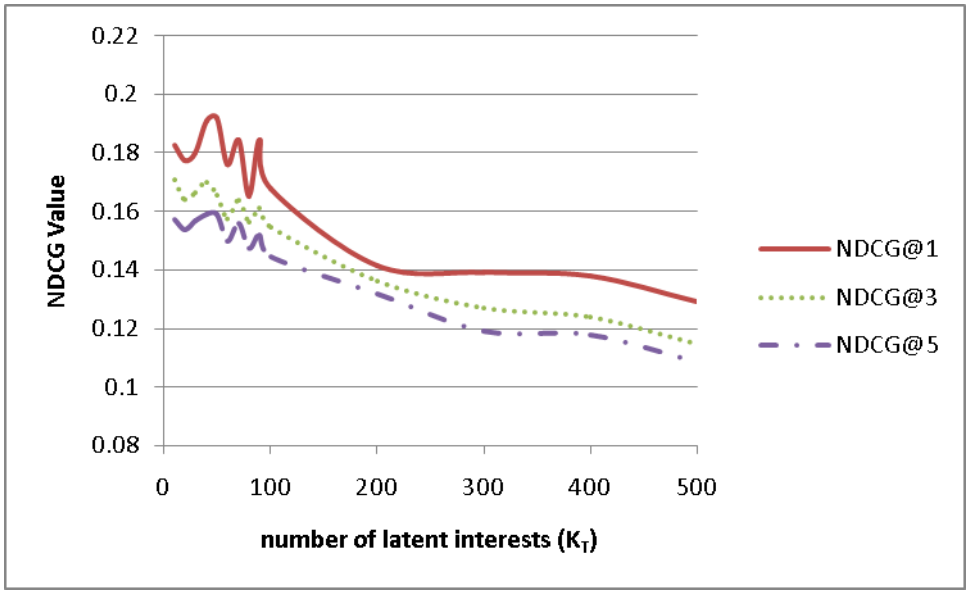

图 4.6 ML1 数据集上 *iter*=1200 时 HTMMF 的 NDCG 表现随 K_T 变化的曲线

图 4.6 和图 4.7 分别描绘了 ML1 和 ML2 数据集上，HTMMF 推荐算法的 NDCG 表现随兴趣主题数 K_T 变化的曲线。实验中，K_T 的取值包括两个区间，分别是 10 到 100 的区间和 100 到 500 的区间。在 10 到 100 区间内，K_T 的取值调整步长为 10，在 100 到 500 区间内，步长为 100。从图中可以发现，针对 NDCG@1、NDCG@3 和 NDCG@5，HTMMF 推荐算法获得最佳效果的 K_T 取值不同。为了更有效的选择参数，我们将 HTMMF 在不同评价方法上的表现进行了归一化。归一化方法如公式(4.22)所示：

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{4.22}$$

[21] 在进行参数设置的实验分析时，HTMMF 算法按照理论设计，在第一阶段使用 BUM 数据模型。

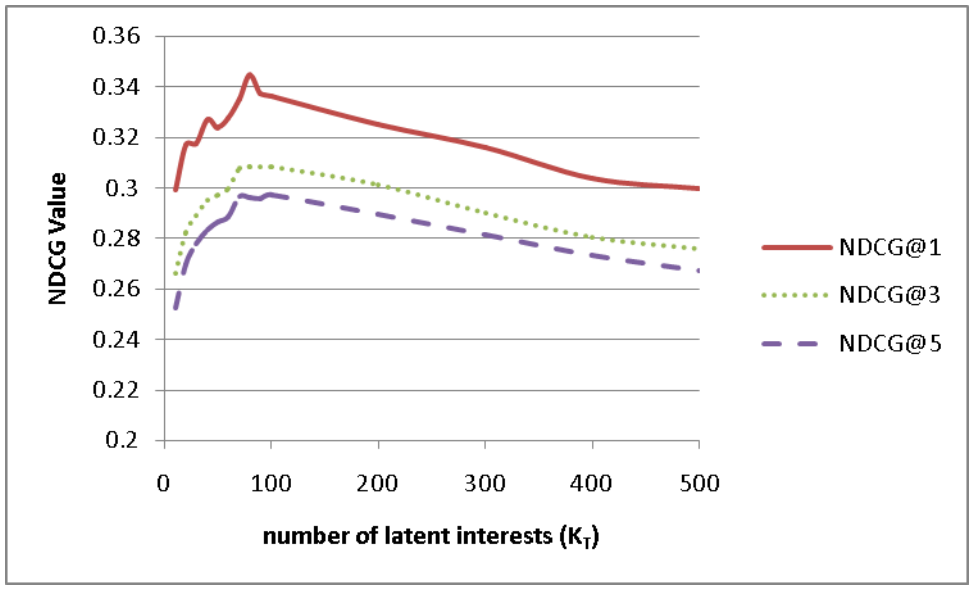

图 4.7 ML2 数据集上 *iter*=1000 时 HTMMF 的 NDCG 表现随 K_T 变化的曲线

其中，X 代表 HTMMF 推荐算法在任意评价方法上的表现，min(X)和 max(X)分别是 HTMMF 推荐算法在该评价方法上的最小值和最大值，X' 为 HTMMF 推荐算法在该评价方法上的归一化表现。

表 4.6 ML1 数据集上 *iter*=1200 时 HTMMF 的归一化表现

K_T	*10*	*40*	*50*	*70*	*80*	*100*	*200*	*300*	*400*	*500*
NDCG@1	0.86	0.99	**1.00**	0.88	0.58	0.61	0.20	0.16	0.14	0.00
NDCG@3	**1.00**	0.99	0.91	0.88	0.75	0.71	0.38	0.22	0.16	0.00
NDCG@5	0.97	1.00	**1.00**	0.94	0.76	0.71	0.46	0.21	0.18	0.00
AVG	0.94	**0.99**	0.97	0.90	0.70	0.68	0.35	0.20	0.16	0.00

表 4.7 ML2 数据集上 *iter*=1000 时 HTMMF 的归一化表现

K_T	*10*	*40*	*50*	*70*	*80*	*100*	*200*	*300*	*400*	*500*
NDCG@1	0.00	0.61	0.54	0.79	**1.00**	0.81	0.57	0.37	0.10	0.01
NDCG@3	0.00	0.69	0.73	0.99	**1.00**	0.99	0.83	0.56	0.34	0.23
NDCG@5	0.00	0.69	0.76	0.98	0.97	**1.00**	0.82	0.65	0.47	0.33
AVG	0.00	0.66	0.68	0.92	**0.99**	0.93	0.74	0.53	0.30	0.19

表 4.6 和表 4.7 分别展示了 ML1 和 ML2 数据集上，HTMMF 算法在各评价方法上的归一化表现。由于页面宽度限制，无法展示所有结果，此处选择了部分 K_T 的典型值进行展示。表中每行是 HTMMF 算法在一种评价方法上的表现，我们加粗显示了

每个评价方法中的最好结果。AVG 是 HTMMF 算法在三个评价方法中的平均表现。从表中可以看出，当 K_T=40 时，HTMMF 算法在 ML1 中表现最好；当 K_T=80 时，HTMMF 算法在 ML2 中表现最好。因此，40 和 80 分别被选作数据集 ML1 和 ML2 中的 K_T 取值，相应的，α 取值分别为 1.25 和 0.625。

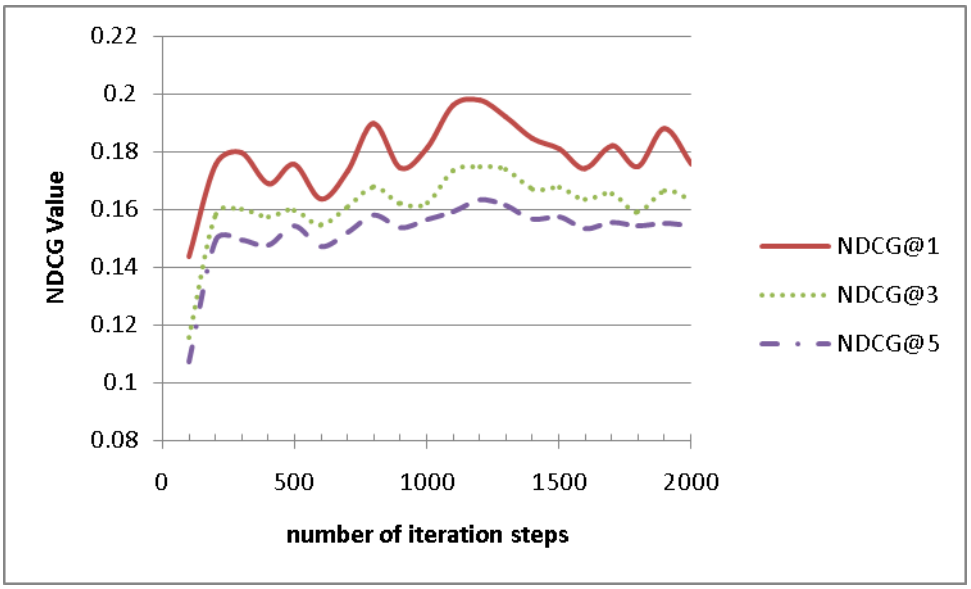

图 4.8 ML1 数据集上 K_T=40 时 HTMMF 的 NDCG 表现随 *iter* 变化的曲线

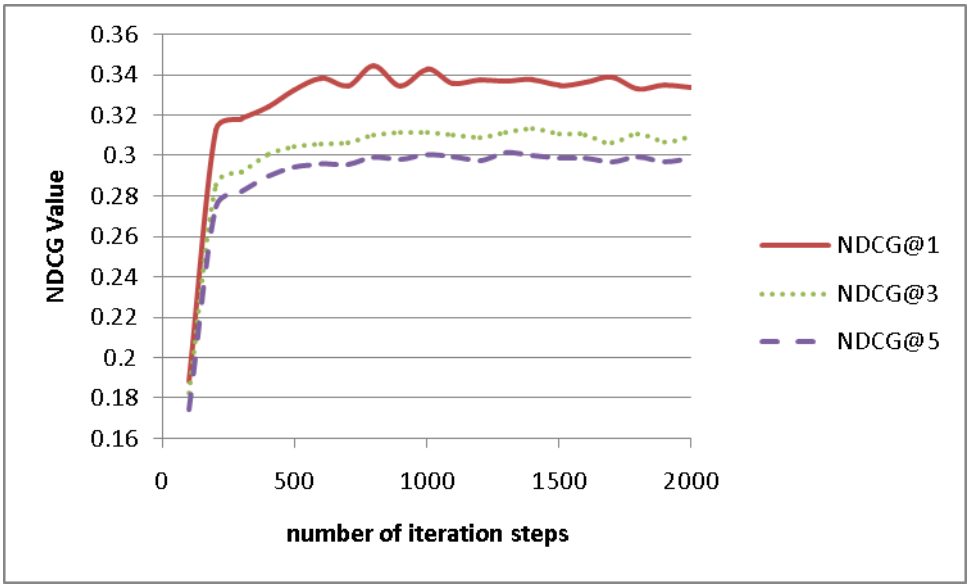

图 4.9 ML2 数据集上 K_T=80 时 HTMMF 的 NDCG 表现随 *iter* 变化的曲线

图 4.8 和图 4.9 分别绘制了 ML1 和 ML2 数据集上，HTMMF 推荐算法的 NDCG 表现随迭代步数 *iter* 变化的曲线。其中，*iter* 的取值是从 100 到 2000 的区间，步长为 100。从图中可以发现，当 *iter* 增加到一定值之后，HTMMF 推荐算法的表现基本稳定。这是因为使用 Gibbs 抽样训练 LDA 模型是一种收敛的方法。在 ML1 数据集中，

HTMMF 推荐算法在 1200 步迭代之后表现最好；在 ML2 数据集中，HTMMF 算法在 1000 步迭代后表现最好。因此，1200 和 1000 分别被选作两个数据集的迭代次数参数。

(2) 第一阶段数据模型的实验分析

依据两步预测推荐算法框架的设计，HTMMF 推荐算法的第一阶段应使用 BUM 数据模型。为了验证该设计的有效性，我们通过实验对比分析 HTMMF 推荐算法在第一阶段分别使用 BUM 模型和 RUM 模型的推荐效果。

表 4.8 ML1 数据集上 HTMMF 在第一阶段使用不同数据模型的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
HTMMF	BUM	**0.2372**	**0.2129**	**0.1979**
	RUM	0.1769	0.1577	0.1470

表 4.9 ML2 数据集上 HTMMF 在第一阶段使用不同数据模型的 NDCG 表现

		NDCG@1	*NDCG@3*	*NDCG@5*
HTMMF	BUM	**0.3377**	**0.3062**	**0.2949**
	RUM	0.3127	0.2900	0.2794

表 4.8 和表 4.9 中分别展示了 ML1 数据集和 ML2 数据集上，HTMMF 推荐算法在第一阶段使用不同数据模型的 NDCG 表现。从表中可以看出，无论是在 ML1 数据集中还是在 ML2 数据集中，HTMMF 推荐算法在第一阶段使用 BUM 数据模型总是比使用 RUM 数据模型具有更好的推荐效果。这就说明两步预测推荐算法框架提出的在第一阶段预测中只使用用户评分行为信息是有效的。

UTCF、ITCF 和 HTMMF 三种两步预测推荐算法在第一阶段使用 BUM 数据模型和使用 RUM 数据模型的对比实验的结果都表明在第一阶段使用 BUM 数据模型的推荐效果更好。这验证了两步预测推荐算法中两阶段数据使用设计的有效性。接下来，我们将对比这三种推荐算法与其它基线算法的推荐效果，验证两步预测推荐算法中两步预测设计的有效性。

4.6.4 两步预测推荐算法与其它推荐算法的比较

本小节将从推荐准确率、多样性等多方面对比分析两步预测推荐算法和基线算法的推荐效果。表 4.10 和表 4.11 分别列出了 ML1 数据集和 ML2 数据集上，这些算法推荐效果的对比结果。表中的每一行是一种算法，每一列是一种评价方法。对于每一

种评价方法，分别加粗了该评价方法下表现最好的算法。

表 4.10 ML1 数据集上不同算法的推荐效果比较

Algorithms		NDCG			Recall			1-Call	COV	CIL	NDCG+
		1	3	5	1	3	5				
Benchmark	UserCF	0.016	0.017	0.018	0.000	0.002	0.003	0.12	**138**	**78**	0.65
	ItemCF	0.007	0.007	0.007	0.000	0.000	0.001	0.06	53	30	0.65
	SVD++	0.034	0.040	0.045	0.001	0.003	0.006	0.21	20	7	**0.71**
	PLPA	0.188	0.156	0.141	0.007	0.015	0.021	0.44	12	0	0.66
	LDA_CF	0.174	0.153	0.142	0.009	0.017	0.027	0.55	129	65	0.69
	iExpand	0.165	0.150	0.141	0.007	0.015	0.021	0.54	126	62	0.66
	PLSA_FP	0.197	0.176	0.166	0.009	0.023	0.032	0.58	57	8	0.65
Proposal	UTCF	**0.270**	**0.235**	**0.219**	**0.012**	**0.028**	**0.040**	**0.65**	68	10	0.65
	ITCF	0.048	0.047	0.046	0.003	0.007	0.011	0.22	56	21	0.69
	HTMMF	0.254	0.218	0.203	**0.012**	0.025	0.036	0.64	101	36	0.63

表 4.11 ML2 数据集上不同算法的推荐效果比较

Algorithms		NDCG			Recall			1-Call	COV	CIL	NDCG+
		1	3	5	1	3	5				
Benchmark	UserCF	0.020	0.021	0.021	0.000	0.001	0.002	0.13	**422**	**277**	0.61
	ItemCF	0.000	0.000	0.000	0.000	0.000	0.000	0.01	71	63	0.79
	SVD++	0.109	0.092	0.087	0.003	0.006	0.009	0.34	43	9	**0.80**
	PLPA	0.178	0.167	0.156	0.004	0.012	0.017	0.53	10	0	0.74
	LDA_CF	0.302	0.267	0.246	0.009	0.020	0.028	0.74	178	51	0.73
	iExpand	0.310	0.287	0.266	0.009	0.022	0.030	0.78	184	54	0.73
	PLSA_FP	0.298	0.272	0.261	0.008	0.020	0.028	0.71	91	5	0.73
Proposal	UTCF	0.319	0.285	0.263	**0.011**	**0.024**	**0.033**	**0.79**	168	39	0.74
	ITCF	0.124	0.120	0.115	0.003	0.009	0.014	0.44	54	15	0.76
	HTMMF	**0.329**	**0.291**	**0.270**	0.009	0.022	0.031	**0.79**	224	80	0.72

观察不同算法在 Top1、Top3 和 Top5 推荐中的 NDCG 效果，可以发现无论是在 ML1 中还是在 ML2 中，UTCF 算法和 HTMMF 算法都是表现最好的算法。它们的良好表现验证了两步预测推荐算法的有效性，也间接验证了 4.2 节中提出的两阶段用户行为模式的有效性。对比 UTCF 算法和 HTMMF 算法，可以发现，在较小的数据集 ML1 中，UTCF 算法效果较好，HTMMF 算法在大数据集 ML2 中表现更好。这表明基于模型的推荐算法更符合大数据的应用背景。PLSA_FP、LDA_CF 和 iExpand 三个算法的 NDCG 表现仅次于 UTCF 和 HTMMF 两种两步预测推荐算法。其中，PLSA_FP

是一种参数级的两步预测推荐算法，其良好表现同样可以验证以两步预测思路解决推荐问题的有效性。此外，LDA_CF 和 iExpand 是两个基于 LDA 模型的推荐算法。它们使用主题模型预测用户对产品感兴趣的概率，并依此向用户推荐产品。这种做法相对于简单的视推荐问题为评分预测问题或排序预测问题更加贴近 Top-*N* 推荐问题的目标，其 Top-*N* 推荐准确率也确实相对更好。ItemCF 算法是所有对比方法中 NDCG 表现最差的算法。这主要是由于 ItemCF 算法本身并不适合对用户进行 Top-*N* 推荐，它更适用的场景是在用户浏览某个产品时，向用户推荐与该产品类似的产品。ITCF 算法是利用两步预测推荐算法框架对传统的 ItemCF 推荐算法的改进。作为一种两步预测推荐算法，ITCF 算法虽然并没有像 UTCF 和 HTMMF 算法一样获得最佳的 NDCG 表现，但是其表现相对于 ItemCF 来说仍有较大的提升。以 ItemCF 为基线，ITCF 算法的 NDCG 表现在 ML1 数据集和 ML2 数据集上分别提升了大约 612%和 28575%。这样的提升进一步验证了两阶段用户行为模式和两步预测推荐算法的有效性。

两阶段用户行为模式和两步预测推荐算法都是建立在用户评分数据是用户主观选择的前提下。而传统的评分预测推荐算法和排序预测推荐算法则是建立在用户随机选择产品进行评分的假设上。因此，若要对它们进行公正的评价，需要一个无论观测数据和缺失数据是否随机产生，都可以利用观测数据对数据全集进行无偏估计的评价方法。有研究表明，Recall 可以完成这样的评价任务[167]。表中第 6、7、8 列是不同算法在 Top1、Top3 和 Top5 推荐中的 Recall 效果。在计算各推荐算法的 Recall 表现时，我们将用户实际评 5 分的产品视作用户喜欢的产品，即令公式(1.18)中的 *L(u)*为测试集中用户 *u* 评 5 分的产品集合。这样的设置可以保证评分预测推荐算法和排序预测推荐算法的 Top-*N* 推荐目标与 Recall 的期望目标相一致[22]。在此评价标准下，UTCF 算法和 HTMMF 算法仍是表现最好的算法，验证了两步预测推荐算法向用户推荐相关产品（即高评分产品）的能力。

表中第 9 列是各算法在 Top5 推荐中的 1-Call 表现。与 NDCG 和 Recall 相似，同样是 UTCF 算法和 HTMMF 算法表现最好，验证了两步预测推荐算法有较好的保证 Top-*N* 推荐中至少有一个相关产品的能力。

综合 NDCG、Recall 和 1-Call 这三种 Top-*N* 推荐准确率的评价结果，UTCF 算法和 HTMMF 算法的表现都优于各基线算法。ITCF 算法的表现虽然并不优越，但其相

[22] 评分预测推荐算法和排序预测推荐算法都是倾向于将用户可能评高分的产品推荐给用户。将 *L(u)*设为用户评 5 分的产品集合使得 Recall 成为评价推荐算法向用户推荐评高分产品的能力。

对于 ItemCF 算法的提升相当可观。这样的结果说明两步预测推荐算法有着很好的推荐准确率，验证了两阶段用户行为模式和两步预测推荐算法的有效性。在几个基线算法中，LDA_CF 和 iExpand 两个算法表现较好，尤其是在较大的 ML2 中，其表现仅次于 UTCF 算法和 HTMMF 算法，要优于参数级的两步预测推荐算法 PLSA_FP。这表明随着数据集的增大，概率生成模型对用户行为的表达能力逐渐变强。

表中第 10、11 列分别是不同算法的 COV 和 CIL 表现。其中，UserCF 表现最好。HTMMF 算法在 ML2 中表现仅次于 UserCF，这说明在大数据情形下，HTMMF 在保证推荐准确率的前提下还可以兼顾推荐结果的多样性和新颖性。UTCF 算法和 ITCF 算法的 COV 和 CIL 表现较差。其主要原因是这两种推荐算法的设计初衷就是倾向于向用户推荐比较热门的产品。UTCF 算法倾向于推荐较多相似用户有过评分行为的产品，ITCF 算法倾向于推荐与用户历史评分产品平均相似度较高的产品。

表中第 12 列是不同算法的 NDCG+表现，该评价方法主要评价各推荐算法对测试集中的产品进行排序的情况。在该评价方法中，SVD++算法表现最好，本章提出的算法表现较差。这说明了两步预测推荐算法对测试集中产品进行排序的效果并不理想。其主要原因是两步预测推荐算法在排序时还考虑了用户是否会对产品进行评分的影响，而这一影响在测试集中并不存在。NDCG+作为一种对比评价方法，推荐算法在该评价方法上的表现好坏并不与其 Top-*N* 推荐效果直接相关。从实验结果可以发现，NDCG 表现好的算法 NDCG+表现未必好，反之亦然。因此，直接利用排序预测优化推荐模型并不一定可以有效提升推荐算法的 Top-*N* 推荐能力。

综上所述，UTCF 和 HTMMF 两种两步预测推荐算法获得了最佳的 Top-*N* 推荐准确率。其表现优于评分预测推荐算法、排序预测推荐算法、基于主题模型的推荐算法以及参数级两步预测推荐算法。此外，HTMMF 算法同时具有良好的推荐多样性和新颖性。这样的实验结果说明，与传统推荐算法相比，两步预测推荐算法可以更好的完成推荐任务，验证了两阶段用户行为模式和两步预测推荐算法的有效性。对比 UTCF 和 HTMMF 算法的推荐准确率，UTCF 在小数据集 ML1 上表现较好，HTMMF 在大数据集上表现更好，说明了基于模型的算法更适用于大数据背景下的推荐任务。

4.7 本章小结

在使用用户显性反馈信息的推荐系统中，无论是评分预测推荐算法还是排序预测推荐算法往往都假设用户是随机选择产品并进行评分的。但是该假设往往并不成立。

这说明传统用户行为模式（即用户随机选择产品并进行评分的行为模式）并不合理。本章针对这个问题，提出了一种两阶段用户行为模式，即在第一阶段中，用户自主选择产品进行评分，第二阶段中，用户对选定产品给出评分值。本章首先通过分析，总结了两阶段用户行为模式和传统用户行为模式的三点主要区别，包括对评分行为信息的重要性的认可程度不同；对观测数据和缺失数据产生原因的观点不同；以及对评分信息，尤其是低评分信息的使用方式不同。针对这些区别，本章使用真实的推荐系统数据集对用户评分信息进行相关性分析，验证了两阶段用户行为模式的有效性。

在两阶段用户行为模式的基础上，本章提出一种对该行为模式进行仿真的两步预测推荐算法框架。该框架分别预测用户对产品评分的概率，以及用户对产品的评分值，然后结合两步预测的结果计算用户对未评分产品评分的数学期望。基于两步预测推荐算法框架，我们提出了两种基于近邻的两步预测推荐算法，包括基于用户的两步预测推荐算法和基于产品的两步预测推荐算法；以及一种基于模型的两步预测推荐算法，该算法利用 LDA 模型预测用户对产品评分的概率，用矩阵分解模型预测用户对产品的评分值。它们都是两步预测推荐算法框架的具体实现。实验结果表明，基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法相对于其它主流推荐算法都有着更好的 Top-*N* 推荐效果，验证了两阶段用户行为模式和两步预测推荐算法的有效性。

第5章 结论

推荐系统是近年来大数据背景下，互联网智能信息处理、数据挖掘等领域的研究热点。它通过分析用户历史行为建立用户兴趣模型，进而向用户推荐符合其兴趣和需求的信息，可以一定程度上解决信息过载问题。推荐系统可以帮助企业捕捉目标用户，提升销售业绩；可以帮助用户扩展兴趣，寻找信息。因此，推荐系统具有较高的科学研究意义和实际应用价值。

协同过滤推荐技术是推荐系统的核心技术之一，得到了学术界和产业界的广泛关注，已经获得了长足的发展并得到了广泛的应用。但是，其中仍存在着诸多亟待解决的问题和挑战。本论文针对其中的流行偏置问题和数据稀疏性问题进行深入研究，并通过解决这些问题，提升了推荐算法在 Top-*N* 推荐中的推荐准确率和多样性。本论文的贡献主要表现在以下三个方面：

1) 针对推荐系统中的流行偏置问题，从减少马太效应对用户模型影响的角度提出一种基于意见的协同过滤推荐算法。该算法利用产品流行度信息和用户意见信息，构建置信度函数，并依此区分不同产品对用户模型的影响，改进基于近邻的协同过滤推荐算法。实验结果表明，基于意见的协同过滤推荐算法既可以提升算法的 Top-*N* 推荐准确率，又可以提升推荐结果对产品的覆盖率，有效缓解了流行偏置问题。

2) 通过对推荐系统中的数据稀疏性问题进行研究，发现缺失数据中存在着用户兴趣的负例信息。为有效利用这些负例信息，本论文分别提出加权法、随机抽样法和近邻抽样法三种对缺失数据建模的方法，然后分别使用这三种方法改进 SVD++算法。改进后的推荐算法可以有效提升推荐结果的准确率和多样性，其改进效果优于其它利用缺失数据的改进方法，亦优于使用排序预测的改进方法。

3) 提出了两阶段用户行为模式和两步预测推荐算法。两阶段用户行为模式解决了传统用户行为模式假设前提不成立的问题，可以更好地描述用户产生评分数据的过程。本论文通过数据分析的形式验证了该行为模式的有效性。然后在此基础上，提出一种对两阶段用户行为模式中用户行为进行仿真的两步预测推荐算法框架。在该框架中，推荐算法分别预测用户对产品评分的概率和具体的评分值，然后整合两步预测的结果向用户进行 Top-*N* 推荐。依据该框架的思想，本论文分别提出了基于近邻的两步预测推荐算法和基于模型的两步预测推荐算法。实验结果表明，两步预测推荐算法相

比主流推荐算法有着更好的 Top-*N* 推荐效果，验证了两阶段用户行为模式和两步预测推荐算法的有效性。

本论文主要对推荐系统中的协同过滤推荐技术的若干方面进行了较深入的研究和探讨，分别针对其中的流行偏置问题和数据稀疏问题，提出了相应的解决方案，提升了算法的 Top-*N* 推荐效果。但是，随着互联网及信息技术的飞速发展，信息过载现象将更加明显，推荐技术也将面临更大的机遇和挑战。我们认为进一步的研究工作可以围绕以下几个方面展开：

1) 领域内推荐技术研究。推荐系统的有效性需要对其拟解决的问题有充分的理解，结合领域特点、有针对性的算法往往比通用算法的推荐效果更好。近年来很多研究者将目标转向了对具体细分领域内的推荐问题进行分析研究，如旅行计划推荐、教育推荐等。在不同的细分领域中，推荐系统面临的问题往往差别很大，例如旅行计划推荐中需要解决时空及成本限制条件下的产品集合推荐问题；教育推荐不仅要向用户推荐其感兴趣的内容，还需向其推荐适合当前学习状态的内容。因此，在后续的工作中我们可以结合待解决问题的领域特点，进行定制化的推荐技术研究。

2) 利用上下文信息的推荐技术研究。本论文介绍的推荐算法主要利用了用户对产品的评分数据，但并未考虑用户所处的上下文信息。典型的推荐系统上下文信息包括时间、地点、用户心情等。不同的上下文条件，用户需要的推荐结果也大不相同，如对分处北京和上海的两个用户推荐餐馆时，选择他们所处城市的餐馆则会较符合用户的期望。这就是不同的地点上下文对推荐结果的影响。在后续的工作中，我们将探讨如何将时间、地点等上下文信息融入推荐技术中，让推荐系统可以更准确的预测用户在特定时刻及特定地点的兴趣。

3) 利用社交信息的推荐技术研究。随着 Facebook，Twitter 等社交网络网站的兴起，社会化过滤已成为了推荐系统的新兴研究领域之一。社会化过滤主要对社交网络中的用户关系进行分析，并依此向用户推荐产品。我们希望在后续的工作中探讨利用社交信息作为用户行为信息的补充和完善，结合协同过滤和社会化过滤的特点的推荐技术，希望这样可以更好的预测用户兴趣，提升用户体验。

参考文献

[1]Rutkowski, A.-F. and C.S. Saunders. Growing Pains with Information Overload[J]. Computer, 2010. 43(6): 96-95.

[2]项亮. 动态推荐系统关键技术研究[D], 2011, 中国科学院自动化研究所.

[3]吴金龙. Netflix Prize 中的协同过滤算法[D], 2010, 北京大学.

[4]项亮. 推荐系统实践[M]. 2012: 人民邮电出版社.

[5]Resnick, P., N. Iacovou, M. Suchak, et al. GroupLens: an open architecture for collaborative filtering of netnews. in Proceedings of the 1994 ACM conference on Computer supported cooperative work[C]. 1994. ACM. 175-186.

[6]Linden, G., B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering[J]. Internet Computing, IEEE, 2003. 7(1): 76-80.

[7]Liu, J., P. Dolan, and E.R. Pedersen. Personalized news recommendation based on click behavior. in Proceedings of the 15th international conference on Intelligent user interfaces[C]. 2010. ACM. 31-40.

[8]Das, A.S., M. Datar, A. Garg, et al. Google news personalization: scalable online collaborative filtering. in Proceedings of the 16th international conference on World Wide Web[C]. 2007. ACM. 271-280.

[9]Bell, R.M. and Y. Koren. Lessons from the Netflix prize challenge[J]. ACM SIGKDD Explorations Newsletter, 2007. 9(2): 75-79.

[10]Shardanand, U. and P. Maes. Social information filtering: algorithms for automating “word of mouth”. in Proceedings of the SIGCHI conference on Human factors in computing systems[C]. 1995. ACM Press/Addison-Wesley Publishing Co. 210-217.

[11]Celma, O. Music recommendation and discovery: The long tail, long fail, and long play in the digital music space[M]. 2010: Springer.

[12]Zheng, H., D. Wang, Q. Zhang, et al. Do clicks measure recommendation relevancy?: an empirical user study. in Proceedings of the fourth ACM conference on Recommender systems[C]. 2010. ACM. 249-252.

[13]Baluja, S., R. Seth, D. Sivakumar, et al. Video suggestion and discovery for youtube: taking random walks through the view graph. in Proceedings of the 17th international conference on World Wide

Web[C]. 2008. ACM. 895-904.

[14]Davidson, J., B. Liebald, J. Liu, et al. The YouTube video recommendation system. in Proceedings of the fourth ACM conference on Recommender systems[C]. 2010. ACM. 293-296.

[15]Kim, J., H. Kim, and J.-h. Ryu. TripTip: a trip planning service with tag-based recommendation. in CHI'09 Extended Abstracts on Human Factors in Computing Systems[C]. 2009. ACM. 3467-3472.

[16]Xie, M., L.V. Lakshmanan, and P.T. Wood. Comprec-trip: A composite recommendation system for travel planning. in Data Engineering (ICDE), 2011 IEEE 27th International Conference on[C]. 2011. IEEE. 1352-1355.

[17]Kurashima, T., T. Iwata, G. Irie, et al. Travel route recommendation using geotags in photo sharing sites. in Proceedings of the 19th ACM international conference on Information and knowledge management[C]. 2010. ACM. 579-588.

[18]Liu, Q., Y. Ge, Z. Li, et al. Personalized travel package recommendation. in Data Mining (ICDM), 2011 IEEE 11th International Conference on[C]. 2011. IEEE. 407-416.

[19]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.

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

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

[22]Bonhard, P. and M. Sasse. ’Knowing me, knowing you’—Using profiles and social networking to improve recommender systems[J]. BT Technology Journal, 2006. 24(3): 84-98.

[23]Krulwich, B. Lifestyle finder: Intelligent user profiling using large-scale demographic data[J]. AI magazine, 1997. 18(2): 37.

[24]Das, M., S. Amer-Yahia, G. Das, et al. Mri: Meaningful interpretations of collaborative ratings[J]. Proceedings of the VLDB Endowment, 2011. 4(11).

[25]Li, Q., Y. Zheng, X. Xie, et al. Mining user similarity based on location history. in Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems[C]. 2008. ACM. 34.

[26]Yang, F. and Z. WANG. A mobile location-based information recommendation system based on GPS

and WEB2. 0 services[J]. database, 2009. 7: 8.

[27]Adomavicius, G. and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques[J]. Knowledge and Data Engineering, IEEE Transactions on, 2012. 24(5): 896-911.

[28]Kohrs, A. and B. Merialdo. Clustering for collaborative filtering applications. in In Computational Intelligence for Modelling, Control & Automation. IOS[C]. 1999. Citeseer.

[29]O’Connor, M. and J. Herlocker. Clustering items for collaborative filtering. in Proceedings of the ACM SIGIR workshop on recommender systems[C]. 1999. Citeseer.

[30]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.

[31]Pavlov, D. and D.M. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. in NIPS[C]. 2002. 1441-1448.

[32]Sarwar, B., G. Karypis, J. Konstan, et al. Item-based collaborative filtering recommendation algorithms. in Proceedings of the 10th international conference on World Wide Web[C]. 2001. ACM. 285-295.

[33]Hofmann, T. and J. Puzicha. Latent class models for collaborative filtering. in IJCAI[C]. 1999. 688-693.

[34]Marlin, B.M. Modeling User Rating Profiles For Collaborative Filtering. in NIPS[C]. 2003.

[35]Koren, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining[C]. 2008. ACM. 426-434.

[36]Li, Y., J. Hu, C. Zhai, et al. Improving one-class collaborative filtering by incorporating rich user information. in Proceedings of the 19th ACM international conference on Information and knowledge management[C]. 2010. ACM. 959-968.

[37]Zhou, Y., D. Wilkinson, R. Schreiber, et al. Large-scale parallel collaborative filtering for the netflix prize[M], in Algorithmic Aspects in Information and Management2008, Springer. 337-348.

[38]Fouss, F., A. Pirotte, J.-M. Renders, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. Knowledge and Data Engineering, IEEE Transactions on, 2007. 19(3): 355-369.

[39]Zhou, T., Z. Kuscsik, J.-G. Liu, et al. Solving the apparent diversity-accuracy dilemma of recommender systems[J]. Proceedings of the National Academy of Sciences, 2010. 107(10): 4511-4515.

[40]Huang, Z., D.D. Zeng, and H. Chen. Analyzing Consumer-Product Graphs: Empirical Findings and

Applications in Recommender Systems[J]. Management Science, 2007. 53(7).

[41]Shang, M.-S., L. Lü, Y.-C. Zhang, et al. Empirical analysis of web-based user-object bipartite networks[J]. EPL (Europhysics Letters), 2010. 90(4): 48006.

[42]Cleger-Tamayo, S., J.M. Fernández-Luna, and J.F. Huete. Top- N news recommendations in digital newspapers[J]. Knowledge-Based Systems, 2012. 27: 180-189.

[43]Pazzani, M.J. and D. Billsus. Content-based recommendation systems[M], in The adaptive web2007, Springer. 325-341.

[44]Baeza-Yates, R. and B. Ribeiro-Neto. Modern information retrieval[M]. Vol. 463. 1999: ACM press New York.

[45]Salton, G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of[M]. 1989: Addison-Wesley.

[46]Song, Y., Z. Zhuang, H. Li, et al. Real-time automatic tag recommendation. in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2008. ACM. 515-522.

[47]Sen, S., J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. in Proceedings of the 18th international conference on World wide web[C]. 2009. ACM. 671-680.

[48]Zhang, Z.-K., C. Liu, Y.-C. Zhang, et al. Solving the cold-start problem in recommender systems with social tags[J]. EPL (Europhysics Letters), 2010. 92(2): 28002.

[49]Ma, H. An experimental study on implicit social recommendation. in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval[C]. 2013. ACM. 73-82.

[50]Lerman, K. Social networks and social information filtering on digg[J]. arXiv preprint cs/0612046, 2006.

[51]Ma, H., H. Yang, M.R. Lyu, et al. Sorec: social recommendation using probabilistic matrix factorization. in Proceedings of the 17th ACM conference on Information and knowledge management[C]. 2008. ACM. 931-940.

[52]Groh, G. and C. Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. in Proceedings of the 2007 international ACM conference on Supporting group work[C]. 2007. ACM. 127-136.

[53]Huang, J., X.-Q. Cheng, H.-W. Shen, et al. Exploring social influence via posterior effect of

word-of-mouth recommendations. in Proceedings of the fifth ACM international conference on Web search and data mining[C]. 2012. ACM. 573-582.

[54]Sinha, R.R. and K. Swearingen. Comparing Recommendations Made by Online Systems and Friends. in DELOS workshop: personalisation and recommender systems in digital libraries[C]. 2001.

[55]Machanavajjhala, A., A. Korolova, and A.D. Sarma. Personalized social recommendations: accurate or private[J]. Proceedings of the VLDB Endowment, 2011. 4(7): 440-450.

[56]Liben‐Nowell, D. and J. Kleinberg. The link‐prediction problem for social networks[J]. Journal of the American society for information science and technology, 2007. 58(7): 1019-1031.

[57]Lo, S. and C. Lin. WMR--A Graph-Based Algorithm for Friend Recommendation. in Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence[C]. 2006. IEEE Computer Society. 121-128.

[58]Akehurst, J., I. Koprinska, K. Yacef, et al. A Hybrid content-collaborative reciprocal recommender for online dating. in International Joint Conference on Artificial Intelligence, IJCAI (in press, 2011)[C]. 2011.

[59]Lancichinetti, A., S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009. 11(3): 033015.

[60]韩家炜, 坎伯. 数据挖掘: 概念与技术[M]. 北京: 机械工业出版社 2001. 232-233.

[61]Agrawal, R., T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. in ACM SIGMOD Record[C]. 1993. ACM. 207-216.

[62]Agrawal, R. and R. Srikant. Fast algorithms for mining association rules. in Proc. 20th int. conf. very large data bases, VLDB[C]. 1994. 487-499.

[63]Han, J., J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. in ACM SIGMOD Record[C]. 2000. ACM. 1-12.

[64]Gedikli, F. and D. Jannach. Neighborhood-restricted mining and weighted application of association rules for recommenders[M], in Web Information Systems Engineering‐WISE 20102010, Springer. 157-165.

[65]Sandvig, J.J., B. Mobasher, and R. Burke. Robustness of collaborative recommendation based on association rule mining. in Proceedings of the 2007 ACM conference on Recommender systems[C]. 2007. ACM. 105-112.

[66]Lin, W., S.A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for

recommender systems[J]. Data mining and knowledge discovery, 2002. 6(1): 83-105.

[67]Uday Kiran, R. and P. Krishna Re. An improved multiple minimum support based approach to mine rare association rules. in Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on[C]. 2009. IEEE. 340-347.

[68]Nakagawa, M. and B. Mobasher. Impact of site characteristics on recommendation models based on association rules and sequential patterns. in Proceedings of the IJCAI[C]. 2003.

[69]Montaner, M., B. López, and J.L. De La Rosa. A taxonomy of recommender agents on the internet[J]. Artificial intelligence review, 2003. 19(4): 285-330.

[70]Adomavicius, G. and A. Tuzhilin. Context-aware recommender systems[M], in Recommender systems handbook2011, Springer. 217-253.

[71]Yuan, Q., G. Cong, Z. Ma, et al. Time-aware point-of-interest recommendation. in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval[C]. 2013. ACM. 363-372.

[72]Burke, R. Hybrid recommender systems: Survey and experiments[J]. User modeling and user-adapted interaction, 2002. 12(4): 331-370.

[73]Chen, T., L. Tang, Q. Liu, et al. Combining factorization model and additive forest for collaborative followee recommendation[J]. KDD CUP, 2012.

[74]夏培勇. 个性化推荐技术中的协同过滤算法研究 [D], 2011, 青岛: 中国海洋大学.

[75]Claypool, M., A. Gokhale, T. Miranda, et al. Combining content-based and collaborative filters in an online newspaper. in Proceedings of ACM SIGIR workshop on recommender systems[C]. 1999. Citeseer.

[76]Pazzani, M.J. A framework for collaborative, content-based and demographic filtering[J]. Artificial Intelligence Review, 1999. 13(5-6): 393-408.

[77]Billsus, D. and M.J. Pazzani. User modeling for adaptive news access[J]. User modeling and user-adapted interaction, 2000. 10(2-3): 147-180.

[78]Tran, T. and R. Cohen. Hybrid recommender systems for electronic commerce. in Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press[C]. 2000.

[79]Smyth, B. and P. Cotter. A personalised TV listings service for the digital TV age[J]. Knowledge-Based Systems, 2000. 13(2–3): 53-59.

[80]Ahmad Wasfi, A.M. Collecting user access patterns for building user profiles and collaborative filtering. in Proceedings of the 4th international conference on Intelligent user interfaces[C]. 1998. ACM. 57-64.

[81]Basu, C., H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. in AAAI/IAAI[C]. 1998. 714-720.

[82]Cremonesi, P., P. Garza, E. Quintarelli, et al. Top-N recommendations on unpopular items with contextual knowledge. in 2011 Workshop on Context-aware Recommender Systems. Chicago[C]. 2011.

[83]Mooney, R.J. and L. Roy. Content-based book recommending using learning for text categorization. in Proceedings of the fifth ACM conference on Digital libraries[C]. 2000. ACM. 195-204.

[84]Sarwar, B.M., J.A. Konstan, A. Borchers, et al. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. in Proceedings of the 1998 ACM conference on Computer supported cooperative work[C]. 1998. ACM. 345-354.

[85]Condliff, M.K., D.D. Lewis, D. Madigan, et al. Bayesian mixed-effects models for recommender systems. in ACM SIGIR[C]. 1999. Citeseer. 23-30.

[86]Schwab, I., A. Kobsa, and I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering[J]. Internal Memo, GMD, St. Augustin, Germany, 2001.

[87]Dietterich, T.G. Machine-learning research[J]. AI magazine, 1997. 18(4): 97.

[88]Delgado, J. and N. Ishii. Memory-based weighted majority prediction. in SIGIR Workshop Recomm. Syst. Citeseer[C]. 1999. Citeseer.

[89]Piotte, M. and M. Chabbert. The pragmatic theory solution to the netflix grand prize[J]. Netflix prize documentation, 2009.

[90]Aksel, F. and A. Birtürk. Enhancing Accuracy of Hybrid Recommender Systems through Adapting the Domain Trends. in Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies (PRSAT 2010)[C]. 2010. 11.

[91]Shang, M.-S., L. Lü, W. Zeng, et al. Relevance is more significant than correlation: Information filtering on sparse data[J]. EPL (Europhysics Letters), 2009. 88(6): 68008.

[92]Herlocker, J.L., J.A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. in Proceedings of the 2000 ACM conference on Computer supported cooperative work[C]. 2000. ACM. 241-250.

[93]Tintarev, N. and J. Masthoff. A survey of explanations in recommender systems. in Data Engineering

Workshop, 2007 IEEE 23rd International Conference on[C]. 2007. IEEE. 801-810.

[94]Lemire, D. and A. Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. in SDM[C]. 2005. SIAM. 1-5.

[95]Wang, P. and H. Ye. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. in Industrial and Information Systems, 2009. IIS'09. International Conference on[C]. 2009. IEEE. 152-154.

[96]Zhang, D. An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing. in Electronic Commerce and Security, 2009. ISECS'09. Second International Symposium on[C]. 2009. IEEE. 215-217.

[97]Shan, H. and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. in Data Mining (ICDM), 2010 IEEE 10th International Conference on[C]. 2010. IEEE. 1025-1030.

[98]Blei, D.M., A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003. 3: 993-1022.

[99]Hofmann, T. Probabilistic latent semantic analysis. in Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence[C]. 1999. Morgan Kaufmann Publishers Inc. 289-296.

[100]Hofmann, T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems (TOIS), 2004. 22(1): 89-115.

[101]Chen, W., W. Hsu, and M.L. Lee. Modeling user's receptiveness over time for recommendation. in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval[C]. 2013. ACM. 373-382.

[102]Paterek, A. Improving regularized singular value decomposition for collaborative filtering. in Proceedings of KDD cup and workshop[C]. 2007. 5-8.

[103]Koren, Y. Factor in the neighbors: Scalable and accurate collaborative filtering[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2010. 4(1): 1.

[104]Sun, J.-T., H.-J. Zeng, H. Liu, et al. Cubesvd: a novel approach to personalized web search. in Proceedings of the 14th international conference on World Wide Web[C]. 2005. ACM. 382-390.

[105]Wu, J. Binomial matrix factorization for discrete collaborative filtering. in Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on[C]. 2009. IEEE. 1046-1051.

[106]Zhang, S., W. Wang, J. Ford, et al. Using singular value decomposition approximation for collaborative filtering. in E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International

Conference on[C]. 2005. IEEE. 257-264.

[107]Rennie, J.D. and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. in Proceedings of the 22nd international conference on Machine learning[C]. 2005. ACM. 713-719.

[108]Zhang, S., W. Wang, J. Ford, et al. Learning from Incomplete Ratings Using Non-negative Matrix Factorization. in SDM[C]. 2006. SIAM.

[109]Hofmann, T. Probabilistic latent semantic indexing. in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval[C]. 1999. ACM. 50-57.

[110]Liu, Q., E. Chen, H. Xiong, et al. Enhancing collaborative filtering by user interest expansion via personalized ranking[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2012. 42(1): 218-233.

[111]Zhang, Y.-C., M. Blattner, and Y.-K. Yu. Heat conduction process on community networks as a recommendation model[J]. arXiv preprint arXiv:0803.2179, 2008.

[112]Zhou, T., J. Ren, M. Medo, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007. 76(4): 046115.

[113]Zhang, Y.-C., M. Medo, J. Ren, et al. Recommendation model based on opinion diffusion[J]. EPL (Europhysics Letters), 2007. 80(6): 68003.

[114]Lü, L. and T. Zhou. Link prediction in complex networks: A survey[J]. Physica A: Statistical Mechanics and its Applications, 2011. 390(6): 1150-1170.

[115]Cremonesi, P., Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. in Proceedings of the fourth ACM conference on Recommender systems[C]. 2010. ACM. 39-46.

[116]Liu, N.N. and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2008. ACM. 83-90.

[117]Liu, N.N., M. Zhao, and Q. Yang. Probabilistic latent preference analysis for collaborative filtering. in Proceedings of the 18th ACM conference on Information and knowledge management[C]. 2009. ACM. 759-766.

[118]Weimer, M., A. Karatzoglou, Q.V. Le, et al. Maximum Margin Matrix Factorization for Collaborative Ranking[J]. Advances in neural information processing systems, 2007.

[119]Rendle, S., C. Freudenthaler, Z. Gantner, et al. BPR: Bayesian personalized ranking from implicit

feedback. in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence[C]. 2009. AUAI Press. 452-461.

[120]Zhang, W., T. Chen, J. Wang, et al. Optimizing top-n collaborative filtering via dynamic negative item sampling. in Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval[C]. 2013. ACM. 785-788.

[121]Shi, Y., A. Karatzoglou, L. Baltrunas, et al. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. in Proceedings of the sixth ACM conference on Recommender systems[C]. 2012. ACM. 139-146.

[122]Pan, R., Y. Zhou, B. Cao, et al. One-class collaborative filtering. in Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on[C]. 2008. IEEE. 502-511.

[123]Pan, W. and L. Chen. Cofiset: Collaborative filtering via learning pairwise preferences over item-sets[J]. training, 2013. 1: 1.

[124]Koren, Y. and J. Sill. OrdRec: an ordinal model for predicting personalized item rating distributions. in Proceedings of the fifth ACM conference on Recommender systems[C]. 2011. ACM. 117-124.

[125]Shi, Y., M. Larson, and A. Hanjalic. Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation[J]. Information Sciences, 2013. 229: 29-39.

[126]Salakhutdinov, R. and A. Mnih. Probabilistic Matrix Factorization. in NIPS[C]. 2007. 2.1.

[127]Shi, Y., M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. in Proceedings of the fourth ACM conference on Recommender systems[C]. 2010. ACM. 269-272.

[128]Hurley, N. and M. Zhang. Novelty and diversity in top-n recommendation--analysis and evaluation[J]. ACM Transactions on Internet Technology (TOIT), 2011. 10(4): 14.

[129]Gunawardana, A. and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks[J]. The Journal of Machine Learning Research, 2009. 10: 2935-2962.

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

[131]Ge, M., C. Delgado-Battenfeld, and D. Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. in Proceedings of the fourth ACM conference on Recommender systems[C]. 2010. ACM. 257-260.

[132]McNee, S.M., J. Riedl, and J.A. Konstan. Being accurate is not enough: how accuracy metrics have

hurt recommender systems. in CHI'06 extended abstracts on Human factors in computing systems[C]. 2006. ACM. 1097-1101.

[133]Celma, Ò. and P. Herrera. A new approach to evaluating novel recommendations. in Proceedings of the 2008 ACM conference on Recommender systems[C]. 2008. ACM. 179-186.

[134]Park, Y.-J. and A. Tuzhilin. The long tail of recommender systems and how to leverage it. in Proceedings of the 2008 ACM conference on Recommender systems[C]. 2008. ACM. 11-18.

[135]Karypis, G. Evaluation of item-based top-n recommendation algorithms. in Proceedings of the tenth international conference on Information and knowledge management[C]. 2001. ACM. 247-254.

[136]Chen, H. and D.R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2006. ACM. 429-436.

[137]Järvelin, K. and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques[J]. ACM Transactions on Information Systems (TOIS), 2002. 20(4): 422-446.

[138]Moffat, A. and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness[J]. ACM Transactions on Information Systems (TOIS), 2008. 27(1): 2.

[139]Zhou, T., R.-Q. Su, R.-R. Liu, et al. Accurate and diverse recommendations via eliminating redundant correlations[J]. New Journal of Physics, 2009. 11(12): 123008.

[140]Newman, M.E. Power laws, Pareto distributions and Zipf's law[J]. Contemporary physics, 2005. 46(5): 323-351.

[141]Soboroff, I. and C. Nicholas. Combining content and collaboration in text filtering. in Proceedings of the IJCAI[C]. 1999. 86-91.

[142]Sarwar, B., G. Karypis, J. Konstan, et al. Application of dimensionality reduction in recommender system-a case study, 2000, DTIC Document.

[143]Quan, T.K., I. Fuyuki, and H. Shinichi. Improving accuracy of recommender system by clustering items based on stability of user similarity. in Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on[C]. 2006. IEEE. 61-61.

[144]Braak, P.t., N. Abdullah, and Y. Xu. Improving the performance of collaborative filtering recommender systems through user profile clustering. in Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on[C]. 2009. IET.

147-150.

[145]Lam, X.N., T. Vu, T.D. Le, et al. Addressing cold-start problem in recommendation systems. in Proceedings of the 2nd international conference on Ubiquitous information management and communication[C]. 2008. ACM. 208-211.

[146]Schein, A.I., A. Popescul, L.H. Ungar, et al. Methods and metrics for cold-start recommendations. in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2002. ACM. 253-260.

[147]Oh, J., S. Park, H. Yu, et al. Novel Recommendation based on Personal Popularity Tendency. in Data Mining (ICDM), 2011 IEEE 11th International Conference on[C]. 2011. IEEE. 507-516.

[148]Lai, S., Y. Liu, H. Gu, et al. Hybrid Recommendation Models for Binary User Preference Prediction Problem[J]. Journal of Machine Learning Research-Proceedings Track, 2012. 18: 137-151.

[149]Sarwar, B., G. Karypis, J. Konstan, et al. Incremental singular value decomposition algorithms for highly scalable recommender systems. in Fifth International Conference on Computer and Information Science[C]. 2002. Citeseer. 27-28.

[150]Ding, Y. and X. Li. Time weight collaborative filtering. in Proceedings of the 14th ACM international conference on Information and knowledge management[C]. 2005. ACM. 485-492.

[151]Koren, Y. Collaborative filtering with temporal dynamics[J]. Communications of the ACM, 2010. 53(4): 89-97.

[152]Lu, Z., D. Agarwal, and I.S. Dhillon. A spatio-temporal approach to collaborative filtering. in Proceedings of the third ACM conference on Recommender systems[C]. 2009. ACM. 13-20.

[153]Xiong, L., X. Chen, T.-K. Huang, et al. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. in SDM[C]. 2010. SIAM. 211-222.

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

[155]Chandramouli, B., J.J. Levandoski, A. Eldawy, et al. StreamRec: a real-time recommender system. in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data[C]. 2011. ACM. 1243-1246.

[156]Smyth, B. and P. McClave. Similarity vs. diversity[M], in Case-Based Reasoning Research and Development2001, Springer. 347-361.

[157]Ziegler, C.-N., S.M. McNee, J.A. Konstan, et al. Improving recommendation lists through topic

diversification. in Proceedings of the 14th international conference on World Wide Web[C]. 2005. ACM. 22-32.

[158]Ishikawa, M., P. Geczy, N. Izumi, et al. Long tail recommender utilizing information diffusion theory. in Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01[C]. 2008. IEEE Computer Society. 785-788.

[159]Kamahara, J., T. Asakawa, S. Shimojo, et al. A community-based recommendation system to reveal unexpected interests. in Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International[C]. 2005. IEEE. 433-438.

[160]Ge, M., F. Gedikli, and D. Jannach. Placing high-diversity items in top-n recommendation lists. in Workshop chairs[C]. 2011. 65.

[161]Onuma, K., H. Tong, and C. Faloutsos. TANGENT: a novel,'Surprise me', recommendation algorithm. in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining[C]. 2009. ACM. 657-666.

[162]Akiyama, T., K. Obara, and M. Tanizaki. Proposal and evaluation of serendipitous recommendation method using general unexpectedness. in Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies (PRSAT 2010)[C]. 2010. 3.

[163]Jin, R., J.Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2004. ACM. 337-344.

[164]Merton, R.K. The Matthew effect in science[J]. Science, 1968. 159(3810): 56-63.

[165]Marlin, B., R.S. Zemel, S. Roweis, et al. Collaborative filtering and the missing at random assumption. in Proceedings of the 23rd conference on Uncertainty in Artificial Intelligence[C]. 2007. 267-275.

[166]Steck, H. Training and testing of recommender systems on data missing not at random. in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining[C]. 2010. ACM. 713-722.

[167]Steck, H. Evaluation of recommendations: rating-prediction and ranking. in Proceedings of the 7th ACM conference on Recommender systems[C]. 2013. ACM. 213-220.

[168]Griffiths, T.L. and M. Steyvers. Finding scientific topics[J]. Proceedings of the National academy of Sciences of the United States of America, 2004. 101(Suppl 1): 5228-5235.

[169]Wei, X. and W.B. Croft. LDA-based document models for ad-hoc retrieval. in Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval[C]. 2006. ACM. 178-185.

攻读博士学位期间发表论文与研究成果清单

[1] 第一作者. Interest before liking: Two-step recommendation approaches. Knowledge-Based Systems, Volume 48, August 2013, Pages 46 – 56. (SCI Indexing, Impact Factor: 4.104, 检索号: 000320893200006).

[2] 第一作者. Opinion-Based Collaborative Filtering to Solve Popularity Bias in Recommender Systems. Database and Expert Systems Applications. Springer Berlin Heidelberg, 2013: 426-433. (EI Indexing, 检索号: 20133916783103).

[3] 第一作者. Considering Rating as Probability Distribution of Attitude in Recommender System. WAIM 2014 Workshop on Big Data Systems and Services. (EI Indexing, Accept).

[4] 第一作者. A hybrid approach of topic model and matrix factorization based on two-step recommendation framework. (Under Review).

[5] 第一作者. Improving Top-N Recommendation Performance Using Missing Data. (Under Review).

[6] 第三作者. A hybrid recommendation algorithm adapted in e-learning environments. World Wide Web, Volume 17, Issue 2, March 2014: 271-284. (SCI Indexing, Impact Factor: 1.196, 检索号: 000330769100006).

攻读博士学位期间参加的科研项目

[1] 面向 Web 文本的属性和属性值知识获取方法研究（国家自然科学基金项目，No：61272361）

[2] Web 文本意见挖掘关键技术研究（国家自然科学基金项目，No：61250010）

[3] 对等网络中基于社区的分布式信息检索方法研究（国家自然科学基金项目，No：61003263）

攻读博士学位期间获奖经历

[1] 2013 年，国家奖学金

[2] 2012~2013 学年，北京理工大学优秀研究生

致谢

前记：2014 年 6 月 9 日，早上的一杯热水，砰！陪伴几载博士生涯的茶杯完成了历史的使命，预示着我学生时代的结束工作征途的伊始。感谢你，感谢你陪我度过的每一天，查论文、看论文、写论文、改论文、写代码、做实验，心情舒畅、心情低落，都有你递上的一杯热水，让我继续奋战；感谢键盘、鼠标、主机、显示器、打印机……，是你们陪我走过一个个的科研日夜。

在本论文即将付梓之际，谨向尊敬的导师牛振东教授致以最真挚的感谢！两年硕士五年博士，是您的热情鼓励和悉心指导，引领我踏上并热爱上这个令我激动的领域。您开阔的学术视野、敏锐的学术洞察力，给予我许多学习、研究上的提点，给我指明了努力的方向，是我取得这微博成绩的重要基石。更重要的是您忘我的工作态度、严谨的治学风格、儒雅的领导风范为我树立了榜样，为我日后的学习、工作点亮了一盏指路的明灯。

感谢计算机学院的领导和老师在我攻读博士学位期间给予我的关心和支持。感谢实验室的陈威老师、刘辉老师、张春霞老师、施重阳老师和金福生老师给予我的热情帮助与勉励。尤其是陈威老师，我们一起做项目、一起搞科研、一起改论文，甚至在您工作调动之后还经常回来对我的学习、研究进行帮助和指导。我的每一粒收获中都倾注着您辛劳的汗水。

感谢团结友爱、奋发向上的实验室团体：赵堃、赵育民、彭学平、江鹏、谷培培、牛科、黄胜、邵琳琳、陈杰、付红萍、刘东磊、于洋、王文韬等。你们是我的良师益友，在学习和生活中给予了我很多帮助和支持。感谢每一个走入我生命的朋友，有你们的陪伴、关心，才有我生活的丰富多彩。

最后也是最重要的，感谢我的父母和家人。你们无私的爱与付出伴随着我出生、成长的每一分每一秒，你们是我最坚定的支持者，是我努力的源动力，是你们的关心、鼓励和帮助才让我克服了学习、工作和生活中的种种困难，一路前行。对于你们，文字无法表达内心的真挚与厚重，只愿我们一起分享和承担未来的一切。

博士生涯即将结束，但所有温暖的记忆都将留至永远。

谨以此文献给所有爱我、关心及帮助我的人们！