

电子科技大学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

博士学位论文

DOCTORAL DISSERTATION



论文题目 时间感知的推荐算法研究

学科专业 计算机软件与理论

学 号 201211060402

作者姓名 李文俊

指导教师 傅彦 教授

分类号 _____ 密级 _____

UDC ^{注1} _____

学 位 论 文

时间感知的推荐算法研究

(题名和副题名)

李文俊

(作者姓名)

指导教师

傅 彦

教 授

电子科技大学

成 都

(姓名、职称、单位名称)

申请学位级别 博士 学科专业 计算机软件与理论

提交论文日期 2017.3.17 论文答辩日期 2017.5.10

学位授予单位和日期 电子科技大学 2017 年 6 月 26 日

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

Research on Time-Aware Recommendation Algorithms

**A Doctoral Dissertation Submitted to
University of Electronic Science and Technology of China**

Discipline: Computer Software and Theory

Author: Li Wenjun

Supervisor: Prof. Fu Yan

School: School of Computer Science and Engineering

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名: _____ 日期: 年 月 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

作者签名: _____ 导师签名: _____

日期: 年 月 日

摘要

随着互联网的高速发展和智能手机的普及，人们已经能够随时随地获取在线内容和服务，例如新闻、音乐、视频、餐厅位置和评论等。这些海量互联网信息在给用户带来便利的同时，也给用户从中筛选感兴趣的内容带来了新的挑战。为了解决这种信息过载问题，推荐系统应运而生。作为连接内容消费者和提供商的桥梁，推荐系统既能帮助消费者发现潜在感兴趣的商品，还能帮助内容提供商提升商品销量和用户粘性。近十年以来，推荐系统在各大商业领域得到了广泛应用，同时也成为计算机、物理、管理学等交叉领域的热点研究课题。

早期研究主要集中在静态推荐系统，即不考虑用户历史行为发生的时间，构建与时间无关的推荐模型。然而人类动力学研究发现，用户的兴趣偏好与时间因素有着紧密联系。例如，用户兴趣会随时间推移发生变化，用户的评分行为存在锚定现象，商品的销量呈现周期性变化等。因此，时间因素的作用近年来吸引了众多推荐系统研究人员的关注。本文基于大量带时间标签的用户行为数据，分析用户动态行为在推荐算法中的作用，对时间感知的推荐算法进行深入研究。论文主要研究工作和创新点如下：

(1) 基于用户显性评价的时间感知算法研究。用户的显性评价通常可以分为正面评价和负面评价两类，不同类型的评价对用户未来的兴趣可能会有不同的影响。本文通过实证分析发现正负面评价具有不同的时间特性，并基于此提出了改进的资源分配算法。该算法可以显著提升推荐结果的准确性，并有效解决了稀疏数据集上的推荐列表个性化问题。为了避免参数遍历寻优的时间开销，本文进一步利用线性回归方法拟合得到了正负面评价的不同时间衰减因子。基于拟合的推荐算法除了能够改善推荐结果之外，还大幅降低了计算复杂度，因此具有较高的商业应用价值。

(2) 基于用户隐性反馈的时间感知算法研究。用户的选择行为一方面可能受到之前购买过该物品的其他用户的影响，另一方面他的选择行为也会影响后续用户。我们针对指定的目标用户，根据选择行为的先后顺序将其他用户分为引领型用户和跟随型用户。实证分析发现跟随型用户选择的物品比较相似，而引领型用户选择的物品个性化程度较高。本文给不同类型用户赋予不同的时间衰减权重，提出了一种基于资源分配的时间感知推荐算法。实验结果表明，相对于时间无关的静态模型，本文算法能够显著提高推荐结果的准确性。

(3) 基于近期流行度的推荐算法研究。在现有推荐算法研究中，物品的流行度

通常定义为它在整个观测周期内被购买的总次数，也就是物品的长期流行度。本文实证分析发现，物品近段时间被购买的次数，也就是物品近期流行度在推荐系统中起着重要的作用。本文通过考虑近期流行度的影响，改进了热传导算法的推荐得分计算公式，并针对资源分配算法设计了推荐结果重排序策略。实验结果表明，上述两种改进算法都能够提供较高的推荐准确性，同时还能提高稀疏数据集上推荐结果的个性化水平。

(4) 基于相对时间的推荐算法研究。在没有明确时间信息的场景中，用户-商品交互记录产生的相对时间同样可以用来提高推荐系统的性能。本文基于不同用户选择同一物品的先后顺序量化不同用户之间的相互影响，并将其作为用户-物品二部图的连边权重，提出了基于用户序列的个性化推荐算法。此外，本文将每个用户表示成她选择过的物品组成的有序概率向量，利用 Jensen-Shannon 散度计算每对用户之间的相似度，提出了基于物品序列的个性化推荐算法。实验结果表明，这两种基于相对时间的推荐算法都可以在保持较高准确性的情况下，显著提高推荐结果的个性化程度。

关键词： 推荐算法，时间感知，相对时间，近期流行度

ABSTRACT

With the rapid development of Internet and the popularity of smart phones, people can get online contents and services whenever and wherever possible, such as news, music, videos, restaurant positions and comments. These huge amounts of Internet information bring users convenience and simultaneously a new challenge of how to filter out the contents of interests. Recommender systems arise to solve this so called “Information Overload” problem. As the bridge connecting content consumers and providers, a recommender system not only helps consumers find the relevant items, but also help providers increase sales and user engagement. In the recent decade, recommender systems were widely employed in the major industrial fields, and receive considerable attentions from the research communities of computer science, physics and management.

Most of early works are focused on static recommender systems, that is, building time-unaware recommendation models regardless of the occurring time of user activities. However, past study on human dynamics found that user behaviors are usually closely related with the time factor. For example, user interests will shift over time, there is a phenomenon of anchoring bias in users rating behaviors, and the sales of products usually suffer periodical fluctuations. In recent years, the effect of time factor in recommendation systems has attracted a lot of research attentions. In this thesis, we investigate the time-aware recommendation algorithms by analyzing the role of user dynamic behaviors, based on a large amount of user-item interaction records with time stamps. The major contributions of this thesis are listed as follow.

(1) The time-aware algorithm based on user explicit ratings. User explicit ratings are usually categorized into positive ratings and negative ratings. Different types of ratings may have different correlations with users future preferences. We empirically discover the different temporal effect of positive and negative ratings, and propose the improved mass-diffusion algorithm by taking into account these temporal effects. This algorithm can significantly improve recommendation accuracy and solve the personalized problem in sparse data sets. In order to save the computation time on traversing the optimal parameters, we make use of linear regression model to fit the different decay factors of positive and negative ratings. This algorithm based on parameter-fitting produces superior recommendation results with lower computation complexity, thus will be of great business value

in future.

(2) The time-aware algorithm based on implicit feedback. The choosing activities of users may be affected by other people who previously selected the same item, and could also affect the subsequent users. Given a specific target user, we categorize other users into two groups, leaders and followers, according to their shopping sequences. The empirical study shows that the shopping lists of followers are more similar to each other, and those of leaders are more personalized. In this thesis, we propose a time-aware diffusion-based recommendation algorithm, which assigns different time-decay weights to different user groups. Experiment results show that our algorithm achieves higher recommendation accuracy, compared with existing time-unaware recommendation models.

(3) The recommendation algorithm based on recent popularity. The item popularity in the existing recommendation models is usually defined by its total purchased times in the whole period of observation, named total popularity of items. Our empirical study finds that the recent popularity of items, defined by the recently purchased times, plays an important role in recommender systems. In this thesis, by taking into account this recent popularity, we improve the heat conduction recommendation algorithms, and propose a resorting method for the recommendation results of mass diffusion model. Experiment results show that, these two recent-popularity based algorithms can achieve good accuracy, and improve the personalized measure in sparse data sets.

(4) The recommendation algorithm based on relative time. In the scenarios without absolute time stamps, the relative time of user-item interaction records is also useful for performance improvement of recommender systems. Based on the collecting sequence of the same items by different users, we quantify the influence among users and accordingly assign weights on the edges of user-item bipartite graph, thus propose a user sequence aware recommendation algorithm. What is more, we model each user as a vector, in which the element is the probability of a permutation on his collected items. The similarity between two users is measured by the Jensen-shannon divergence. Based on this, we proposed an item sequence aware recommendation algorithm. Experiment results demonstrate that these two methods can both improve the diversity, novelty and coverage while keeping the high accuracy.

Keywords: recommender algorithm, time awareness, relative time, recent popularity

目 录

| | |
|-----------------------------------|-----------|
| 第一章 绪论..... | 1 |
| 1.1 研究工作的背景与意义 | 1 |
| 1.2 国内外研究现状..... | 3 |
| 1.2.1 人类行为的时间特性研究 | 3 |
| 1.2.2 静态推荐算法研究 | 5 |
| 1.2.3 时间感知推荐算法研究..... | 7 |
| 1.3 本文的主要研究内容与组织架构..... | 9 |
| 第二章 推荐系统基础知识 | 12 |
| 2.1 推荐系统的商业应用 | 12 |
| 2.2 推荐系统面临的主要挑战 | 16 |
| 2.3 常用的推荐算法..... | 19 |
| 2.3.1 协同过滤推荐算法 | 19 |
| 2.3.1.1 基于内存的协同过滤算法..... | 19 |
| 2.3.1.2 基于模型的协同过滤算法..... | 20 |
| 2.3.2 基于内容的推荐算法 | 21 |
| 2.3.3 基于上下文的推荐算法..... | 22 |
| 2.3.4 基于网络结构的推荐算法 | 22 |
| 2.3.5 混合推荐算法 | 24 |
| 2.4 推荐系统评测指标 | 25 |
| 2.4.1 准确性指标..... | 25 |
| 2.4.2 个性化指标..... | 27 |
| 2.5 实验数据集介绍..... | 28 |
| 2.5.1 公开数据集..... | 28 |
| 2.5.2 数据集划分方式..... | 29 |
| 2.6 本章小结 | 31 |
| 第三章 基于显性评分的时间感知推荐算法研究..... | 32 |
| 3.1 问题的提出 | 32 |
| 3.2 用户显性行为的时间特性分析 | 33 |
| 3.2.1 不同用户的评分标准差异 | 33 |
| 3.2.2 正负面评价的时间特性差异 | 35 |

| | |
|-----------------------------------|-----------|
| 3.3 正负面评价时序特性在推荐算法中的应用研究 | 37 |
| 3.3.1 基于正负面评价的时间感知推荐算法 | 38 |
| 3.3.2 基于曲线拟合的时间感知推荐算法..... | 39 |
| 3.4 算法性能评测 | 41 |
| 3.4.1 实验数据集和对比算法..... | 41 |
| 3.4.2 基于正负面评价的时间感知推荐算法性能分析..... | 42 |
| 3.4.3 不同算法的推荐性能比较 | 45 |
| 3.4.4 基于曲线拟合的时间感知推荐算法性能分析 | 47 |
| 3.5 本章小结 | 48 |
| 第四章 基于隐性反馈的时间感知推荐算法研究..... | 50 |
| 4.1 问题的提出 | 50 |
| 4.2 用户隐性反馈行为的时间特性分析 | 50 |
| 4.2.1 用户个体行为的时间特性 | 50 |
| 4.2.2 引领型邻居和跟随型邻居 | 52 |
| 4.3 基于资源分配的时间感知推荐模型 | 53 |
| 4.3.1 用户之间的时序影响 | 54 |
| 4.3.2 用户个体的兴趣漂移 | 55 |
| 4.4 算法性能评测 | 56 |
| 4.4.1 实验数据集和对比算法..... | 56 |
| 4.4.2 时间感知算法的性能分析 | 57 |
| 4.4.3 不同算法的实验结果对比 | 59 |
| 4.5 本章小结 | 61 |
| 第五章 基于物品近期流行度的推荐算法研究 | 62 |
| 5.1 问题的提出 | 62 |
| 5.2 物品的时间特性研究 | 62 |
| 5.3 基于近期流行度的推荐算法..... | 64 |
| 5.3.1 物品的近期流行度 | 64 |
| 5.3.2 融合近期流行度的推荐算法 | 65 |
| 5.3.3 融合近期流行度的推荐算法性能分析 | 66 |
| 5.4 近期流行度在热传导推荐算法中的研究..... | 68 |
| 5.4.1 基于近期流行度的热传导算法 | 68 |
| 5.4.2 基于近期流行度的热传导算法性能分析 | 69 |
| 5.5 基于近期流行度的后处理算法 | 73 |

| | |
|--------------------------------|------------|
| 5.5.1 后处理算法在资源分配算法中的应用 | 73 |
| 5.5.2 后处理算法的性能分析..... | 74 |
| 5.6 本章小结 | 76 |
| 第六章 基于相对时间的推荐算法研究 | 78 |
| 6.1 问题的提出 | 78 |
| 6.2 基于用户序列的资源分配算法 | 79 |
| 6.2.1 相关性计算..... | 79 |
| 6.2.2 偏好预测 | 80 |
| 6.3 基于物品序列的资源分配算法 | 80 |
| 6.3.1 相似度计算..... | 81 |
| 6.3.2 偏好预测 | 83 |
| 6.4 算法性能评测 | 83 |
| 6.4.1 实验数据集和对比算法..... | 83 |
| 6.4.2 基于用户序列的资源分配算法性能分析 | 84 |
| 6.4.3 基于物品序列的资源分配算法性能分析 | 86 |
| 6.4.4 不同算法的推荐性能对比 | 87 |
| 6.5 本章小结 | 89 |
| 第七章 总结与展望 | 90 |
| 7.1 研究工作总结 | 90 |
| 7.2 未来工作展望 | 91 |
| 致 谢 | 92 |
| 参考文献 | 94 |
| 攻读博士学位期间取得的成果 | 107 |

图目录

| | |
|---|----|
| 图 1-1 阿里云推荐系统架构图 | 2 |
| 图 1-2 不同时间用户签到行为的相似度 | 4 |
| 图 1-3 用户评分行为的锚定偏向..... | 5 |
| 图 1-4 论文的章节关系 | 10 |
| 图 2-1 京东商城商品推荐界面 | 13 |
| 图 2-2 QQ 音乐客户端个性化电台推荐界面..... | 14 |
| 图 2-3 今日头条首页界面 | 15 |
| 图 2-4 二部图上资源分配算法示意图 | 24 |
| 图 2-5 K 折交叉验证示意图 | 30 |
| 图 2-6 动态推荐算法的数据集划分方式示意图 | 31 |
| 图 3-1 淘宝和豆瓣的评分系统界面 | 32 |
| 图 3-2 用户评分标准差的分布情况。 | 34 |
| 图 3-3 用户在第一个月和最后一个月平均评分的对应关系。 | 35 |
| 图 3-4 会话中正面评价物和负面评价关联的物品组成的集合序列 | 36 |
| 图 3-5 (a) MovieLens、(b) Netflix 和 (c) MovieTweetings 数据集上的 $CorP$ 、 $CorN$ 、 $CorP - R$ 和 $CorN - R$ 曲线..... | 37 |
| 图 3-6 时间段图模型样例 | 37 |
| 图 3-7 TaPNR 算法 F1 指标与参数 λ 和 μ 的关系。 | 43 |
| 图 3-8 历史评价与未来偏好间的相似度随时间间隔的变化关系。 | 44 |
| 图 3-9 TaPNR 推荐准确性与参数 β 的关系。 | 45 |
| 图 3-10 TaPNR 推荐准确性与参数 η 的关系。 | 46 |
| 图 4-1 (a) Netflix、(b) FriendFeed 和 (c) MovieLens 数据集上 SoT、SoS 和 SoR 的统计结果 | 52 |
| 图 4-2 基于资源分配的时间感知推荐算法示意图 | 55 |
| 图 4-3 不同推荐算法随参数变化的推荐准确性情况。 | 58 |
| 图 4-4 不同算法的推荐准确率与训练集时间长度的关系。 | 59 |
| 图 5-1 MovieLens 数据集中物品流行度的分布情况 | 63 |
| 图 5-2 物品长期流行度与平均在线天数之间的关系 | 63 |

| | |
|---|----|
| 图 5-3 “Pink Floyd: The Wall” 和 “Insomnia” 评分人数随时间变化情况 .. | 64 |
| 图 5-4 $(t^* - T_P, t^*)$ 和 $(t^*, t^* + T_F]$ 间物品的度相关性 | 65 |
| 图 5-5 皮尔逊相关性与时间窗口 T_P 的关系 | 65 |
| 图 5-6 MovieLens 数据集上 HP-MD 和 HP-HC 算法的推荐性能 | 68 |
| 图 5-7 Netflix 数据集上 HP-MD 和 HP-HC 算法的推荐性能 | 68 |
| 图 5-8 MovieLens 数据集上 HP-HPH、HP-GHC 和 HP-DWC 算法的推荐 性能 | 70 |
| 图 5-9 Netflix 数据集上 HP-HPH、HP-GHC 和 HP-DWC 算法的推荐性能 ... | 71 |
| 图 5-10 MovieLens 数据集不同用户组的推荐准确率 | 73 |
| 图 5-11 Netflix 数据集不同用户组的推荐准确率 | 73 |
| 图 5-12 MovieLens 和 Netflix 数据集上后过滤处理的推荐性能..... | 75 |
| 图 5-13 不同算法的推荐准确率与训练集时间长度的关系。 | 76 |
| 图 6-1 用户评分偏好顺序样例 | 78 |
| 图 6-2 用户序列样例..... | 80 |
| 图 6-3 USD 算法中参数对推荐性能的影响 | 85 |
| 图 6-4 NBI 和 USD 算法推荐列表总物品的度分布。 | 86 |
| 图 6-5 NBI 和 ISD 算法推荐列表总物品的度分布。 | 87 |

表目录

| | |
|--|----|
| 表 3-1 符号和标记列表 | 34 |
| 表 3-2 第一组数据集基本统计信息 | 41 |
| 表 3-3 第二组数据集基本统计信息 | 42 |
| 表 3-4 对比算法说明..... | 43 |
| 表 3-5 TaPNR 算法考虑和不考虑时间因素时的推荐结果 | 44 |
| 表 3-6 MovieLens 数据集上不同算法的推荐性能对比..... | 46 |
| 表 3-7 Netflix 数据集上不同算法的推荐性能对比..... | 47 |
| 表 3-8 MovieTweetings 数据集上不同算法的推荐性能对比 | 47 |
| 表 3-9 CorP 和 CorN 曲线拟合结果 | 48 |
| 表 3-10 MovieLens、Netflix 和 MovieTweetings 数据集上 TaCF 算法的推 荐性能 | 48 |
| 表 3-11 MovieLens 数据集上 TaPNR 和 TaCF 算法性能提升百分比对比结果 | 48 |
| 表 3-12 Netflix 数据集上 TaPNR 和 TaCF 算法性能提升百分比对比结果..... | 49 |
| 表 3-13 MovieTweetings 数据集上 TaPNR 和 TaCF 算法性能提升百分比对 比结果 | 49 |
| 表 4-1 Netflix、FriendFeed 和 MovieLens 数据集上 SoG-L 和 SoG-F 统计 结果 | 53 |
| 表 4-2 数据集基本统计信息 | 57 |
| 表 4-3 时间因素对 TaDb 算法推荐性能的影响 | 58 |
| 表 4-4 TaDb 相比于其他算法在准确率方面提高的百分比 | 60 |
| 表 4-5 对比算法的推荐多样性比较 | 60 |
| 表 4-6 对比算法的推荐新颖性比较 | 61 |
| 表 5-1 数据集基本统计信息 | 67 |
| 表 5-2 MovieLens 和 Netflix 数据集上不同算法的性能比较..... | 72 |
| 表 6-1 数据集基本统计信息 | 83 |
| 表 6-2 MovieLens 数据集上 USD 和 NBI 算法的推荐结果对比 | 84 |
| 表 6-3 MovieTweetings 数据集上 USD 和 NBI 算法的推荐结果对比..... | 86 |
| 表 6-4 参数 k 对基于物品序列的资源分配算法的影响 | 87 |

表目录

| | |
|---|----|
| 表 6-5 MovieLens 数据集上各算法的推荐结果对比 | 88 |
| 表 6-6 MovieTweetings 数据集上各算法的推荐结果对比 | 88 |

第一章 绪论

1.1 研究工作的背景与意义

随着移动互联网的普及和共享经济的发展^[1,2]，用户获取电子信息和网上服务的渠道越来越便捷、成本越来越低。同时，由于创业成本的降低和政府政策的鼓励，互联网上涌现出越来越多的内容提供商。大量的有用信息和垃圾数据同时涌入到互联网中，用户面对的信息数量呈指数级的增长^[3,4]。例如，喜马拉雅 FM 平台目前已经有超过 3000 万条有声内容，覆盖财经、音乐、新闻、商业、小说、汽车等 328 个门类^[5]；每天有 2 亿多条动态发布在新浪微博上^[6]；Amazon 拥有数百万本图书资源^[7]；淘宝网拥有百万级别的商家，这些商家提供了数以亿计的产品^[8]。丰富的信息资源给人们带来便利的同时，也让人们忍受着无法从海量数据中找到有用信息的困扰。用户面对如此庞大規模的数据时已经无法通过个人的能力对其进行筛选和过滤，这种现象被称为信息过载（information overload）^[9,10]。在这个大数据时代，作为商品提供者的企业也同样面临着信息过载的问题。如何在同行中脱颖而出，将自己的内容推送到合适的目标用户面前已经成为各企业急需攻克的难题。

搜索引擎是解决信息过载的一种重要工具，其要求用户使用关键词作为需求的描述信息，系统根据关键词为用户返回最相关的结果，同时系统一定程度上对垃圾内容进行了过滤。搜索引擎是一种被动式的信息过滤技术，需要用户能够对自己的需求选择合适的关键词进行描述^[11]。但是大多数时候，用户对自己的需求是不明确的，或者由于能力的不足找不到合适的关键词。面对这种情况，研究人员发明了推荐系统这种主动式的信息过滤工具。推荐系统可以被定义为：通过基于物品、用户和两者间交互行为的相关信息来预测用户对特定物品的喜爱程度，以尝试向特定用户（个人或企业）推荐最合适的物品（产品或服务）的程序^[12-14]。推荐系统不需要用户对自己的需求精心挑选合适的关键词，它通过分析用户之间历史行为的相似度来发现与目标用户相似的人群，通过相似人群的兴趣来预测目标用户对特定物品的偏好^[14]。在如今这个快速发展的互联网社会中，搜索引擎和推荐系统对于用户来说都是不可或缺的信息过滤手段。

图1-1是阿里云提供的在线推荐系统架构图^[15]，主要包括用户和物品的数据收集模块、用户偏好统计分析模块、推荐算法模块和 UI 展示模块等。在这些模块中，推荐算法是推荐系统的核心，也是众多科学研究工作关注的重点。经典的推荐算法包括基于内容的推荐算法（分析物品之间的相似性来对用户偏好进行建模）、协

同过滤算法（研究相似用户行为）和两种方式的混合算法（将两种类型的算法进行组合以克服各自的缺陷）。目前，这些推荐算法已经成功应用于众多商业网站中并且产生了巨大的商业价值^[16]。DVD 租赁提供商 Netflix 在用户浏览电影信息时，会为其展示每部电影的评分信息，以帮助用户决定租借哪部电影^[17]；在线图书零售商 Amazon 提供了每本图书的平均用户评分，并且在用户购买一本图书后会向其展示购买该本图书的其他用户购买的图书清单^[18]；Microsoft 为用户提供免费的软件下载服务，当用户下载了其中一些软件时，系统会推荐用户下载其他的相关软件。随着互联网的应用场景不断增加，科研人员又开发了许多新的推荐算法，包括基于社交网络的推荐算法^[19,20]、模糊推荐算法^[21,22]，基于上下文感知的推荐算法^[23] 和群组推荐算法^[24,25] 等。

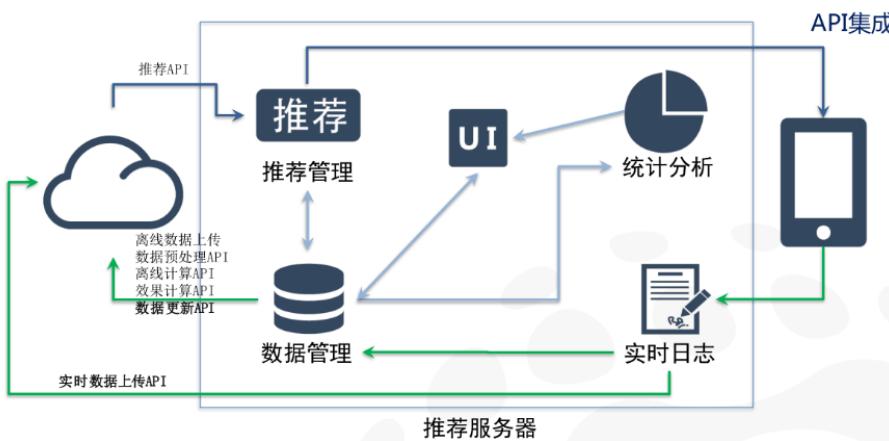


图 1-1 阿里云推荐系统架构图^[15]

虽然推荐系统在商业领域取得了巨大的成功，但是依然存在许多问题值得深入研究。早期的推荐系统主要关注于用户静态行为的分析，这是因为当时对人类行为的时空动力学研究还很少涉及，同时，推荐技术在静态系统中的研究也不够充分。随着近年来对人类时空行为的深入研究^[26-28]，研究人员发现挖掘用户的行为模式能够帮助推荐系统提高推荐效果并使其应用于更为复杂的场景中。例如，新用户和老用户的行为往往具有很大的差异性。由于新用户缺少足够的领域知识，通常情况下更倾向于选择流行的商品；而老用户往往形成了自己独特的兴趣爱好，对小众商品的关注要明显高于新用户^[29]。这种现象导致新用户选择的商品更为集中，老用户选择了更多的长尾商品^[30]。

人类行为模式的统计特性可以分为空间特性和时间特性。随着移动智能手机的普及，用户的 GPS 信息为研究人类行为的空间特性和开发基于位置的推荐系统^[31,32] 提供了丰富的数据支持。例如，通过分析用户每天驾车的导航信息可以推断出用户的工作地、居住地和平时常去的商业圈等。用户出现在陌生地点时，系统

可以首先分析用户需要进餐的可能性然后为其推荐附近的餐厅。将位置信息作为附加的上下文标签可以在广告推荐^[33] 和朋友推荐^[34] 等领域提高推荐算法的精度。此外，推荐系统还可以基于用户的位置信息计算用户间或地点间的相似度，缓解推荐系统数据稀疏的问题。

时间特性是人类行为模式中又一个值得深入探索的领域。自从 Barabási 在 2005 年揭示了人类行为的时间间隔不再是以前认为的泊松过程而是一种幂率分布后，针对人类行为时间特性的探索吸引了众多研究学者的关注^[35]。在推荐系统中，随时都会有新的用户和新的物品加入进来，同时，用户与物品的交互行为也在不断增加，整个系统是在不断变化中的。时间信息不仅可以帮助系统跟踪用户偏好的演化过程^[36]，识别用户习惯和兴趣的周期性特征^[37]，还能作为上下文感知系统的输入条件^[38]。例如，某用户习惯在 9 点到 10 点之间阅读新闻资讯内容（很可能是上班途中），后台程序在 8 点 50 分向用户推送其可能感兴趣的新闻会使用户得到很好的体验，而在 10 点以后再推送就显得不明智了。从时间相关的数据中还能分析出用户的长期和短期兴趣，通过将两者分离可以显著提高推荐系统的性能^[39,40]。

由此可见，通过时间信息分析用户的在线行为，可以更好的理解用户的偏好，发现用户之间的相互影响，帮助研究人员设计更合适的推荐算法，解决更加复杂场景下的推荐问题。这对众多的内容提供商有着重要的商业价值。

1.2 国内外研究现状

1.2.1 人类行为的时间特性研究

人类社会的一个重要特征是每个人都有着许多不同的兴趣爱好。有的兴趣爱好可以伴随用户一生，而有些会在短时间内消失。个人兴趣的改变是许多人类行为的背后驱动力。用户会因为兴趣的消失而退出某个社团，会因为突然的兴趣加入新的系统；兴趣的加深使得用户购买和挑选更多关于该领域的商品，兴趣的降低使得用户选择相应商品的频率随之降低。人类的这些行为使得线上或者线下系统都处于不断变化的状态中。人类行为的动态特性对商业、医疗、乃至国防行业都有着重要的研究意义。尤其是在商业领域，充分了解用户兴趣随时间变化的特征是企业成功的关键，这些知识对于广告投放和产品推广有着巨大的商业价值。

目前，已经有很多研究工作对社交领域和经济领域的人类行为开展了分析和建模工作，相关的实证结论被应用到资源分配、运输控制、流行病预测和个性化推荐系统中^[26,41,42]。依靠信息技术和大量互联网数据的帮助^[43]，研究人类行为的模式规律变得越来越方便。最近，研究结果表明人类行为的时间间隔分布并非服从泊松分布，而是呈现幂率特征^[44-48]。这种非泊松分布的特性意味着，用户在较

长时间内的沉默后通常会出现短时间内突然的高频率行为。研究人员试图通过许多机制来说明这种现象，例如最高优先级队列模型^[35,49]，泊松概率模型^[50,51]，用户兴趣的变化模式^[52]，记忆性作用^[53]和人类交互行为^[48,54,55]等。

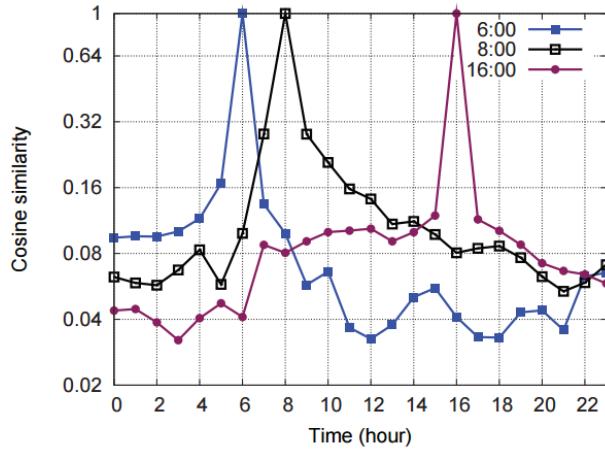
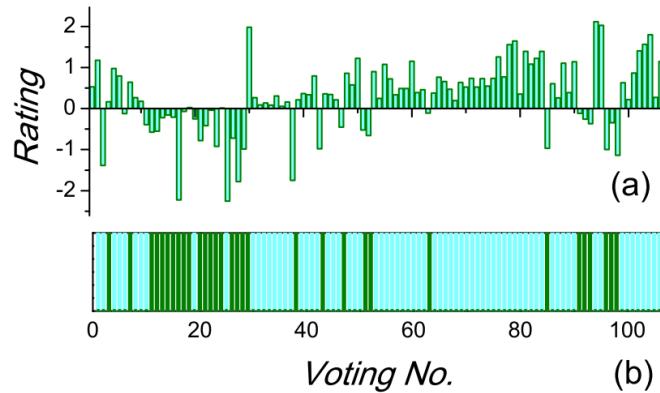


图 1-2 不同时间用户签到行为的相似度^[56]

相对人类行为的时间间隔和轨迹动力学特性，研究人类行为背后的兴趣演化特性要困难得多，一方面是因为没有合适的方式用来对兴趣进行量化，另一方面是由于能够用来推断用户兴趣的数据还相对欠缺。近年来，由于电子商务和移动通讯的快速发展，用户频繁的消费行为为研究机构提供了超大规模的研究数据集，这为量化用户兴趣和推断其中的动力学特性提供了可能。作为大数据学科的分支，人类兴趣动力学领域的研究工作才刚刚开始它的脚步。Zhang 等人^[57]对 Foursquare 和 Gowalla 中用户签到地点的时序相关性进行了大量实证分析，发现相似用户拜访相同地方的时间间隔一般分布在 3 到 6 个小时内，同一用户每隔四个小时会拜访一次相似的地方，相似用户拜访同一个地方的时间都集中在同一个时间段内。Zhao 等人^[58]通过对豆瓣和淘宝数据集的分析，发现人类兴趣动力学背后存在三个基本的原因，分别是用户会优先根据历史兴趣做出行为选择、行为具有惯性作用和用户喜欢对新兴趣进行探索。如图 1-2 所示，Yuan 等人^[56]对用户在社交软件中的签到信息进行了分析，根据用户-地点-时间的三维数据度量了不同时间用户签到行为的相似度，发现距离相近的时间段内用户签到的地点十分相似，时间隔得越远，签到地点的差异性越大。Yang 等人^[59]从用户的在线电影评分行为中发现用户的评分行为存在明显的锚定偏向（如图 1-3 所示），即用户当前的打分行为会以最近的某次打分作为参考点，并非只受当前物品的质量决定。

图 1-3 用户评分行为的锚定偏向^[59]

1.2.2 静态推荐算法研究

推荐问题是一个依赖于评分信息的概念，评分信息作为一种表达用户对不同物品的偏好程度的信息被应用于众多领域中^[60]。用 U 表示用户集合， O 表示物品集合。静态推荐系统可以表述成一个兴趣函数 F ，用以计算用户 $u_i \in U$ 将对物品 $o_a \in O$ 打的分数 r_{ia} 的预测值 \hat{r}_{ia} ：

$$F: U \times O \rightarrow R \quad (1-1)$$

其中， R 表示用户可用的评分空间，例如正整数或者特定范围内的实数等。

推荐系统的核心价值是长尾商品的曝光和销售。《连线》杂志主编 Chris Anderson，被誉为“信息时代的精神领袖”，于 2004 年提出了长尾理论^[61]。他发现在如今的互联网上，先进的网络技术使得用户获取信息的成本变得越来越低，一些早前被认为是冷门的产品在互联网上逐渐受到越来越多用户的关注。冷门物品的持续曝光会给在线网站带来巨大的商业价值。以 Amazon 网络书店与传统大型书店为例，市场上每年发行出版的书籍超过数百万种，但是受场地空间的约束，传统书店无法上架所有的图书，并且能够摆放在书店畅销书架上的图书更是少之又少。因此传统书店以销售畅销书作为主要的经营模式。但是，Amazon 等网络书店却不受场地的限制，用户有更多的机会接触和浏览长尾书籍，这就为长尾书籍的销售提供了更大的空间。虽然小众书籍每本的销售量并不大，但是因为这类书籍的种类比热门书籍要多得多，因此小众书籍累积的销售额反而会超过畅销书的累积销售额。

推荐系统的研究源于信息检索和信息过滤^[62]。20 世纪 90 年代中期，研究人员开始关注于如何利用用户对物品的评分信息产生推荐结果，至此之后，推荐系统发展成为学术界一门独立的学科^[63]。美国明尼苏达大学的 GroupLens 研究小组

是最早进入该领域的研究团队，并取得了辉煌的成就。常用的静态推荐技术包括协同过滤算法（Collaborative Filtering，简称 CF）^[64]，基于内容的（Content-based，简称 CB）推荐算法^[65]、基于知识的（knowledge-based，简称 KB）推荐算法^[66] 和混合推荐算法等。

协同过滤算法是学术研究和商业应用中使用最普遍的一种推荐算法。它的基本假设是：在过去具有相似兴趣的人更可能在将来也具有相同的兴趣。经典的协同过滤算法首先分析用户过去的行为找到与目标用户相似的邻居，然后利用邻居对物品的评分信息产生推荐结果。协同过滤算法能够准确的推荐音乐和电影这类复杂的商品，即使是在无法得知商品具体内容的情况下。例如，在测量物品间或用户间的相似度时，可以使用 k 最近邻^[67] 或 Pearson 相关性^[68] 这两个流行的相似度计算模型。它们不需要知道诸如用户的人口统计学属性和物品的具体属性，只依赖于用户与物品的历史交互信息（即哪些用户共同购买了这些物品并且对它们的评分是多少）。协同过滤算法可以分为基于内存和基于模型的两类算法。基于内存的协同过滤算法使用数据库存储用户和物品的交互信息，以此计算物品间或用户间的相似度并保存在内存中，最后根据相似度信息挑选出 K 个最近邻居进行评分预测。该算法根据相似度的主体不同可以分为基于用户的^[69] 和基于物品的^[18, 70, 71] 协同过滤算法。在物品数量和类型相对稳定的系统中，基于物品的协同过滤算法比基于用户的算法能够得到更好的推荐结果，并且提供很好的推荐解释。基于模型的协同过滤算法直接通过评分数据来学习用户的偏好模型，不再对用户间和物品间的相似度进行计算和存储。核映射推荐算法^[72]、贝叶斯分类器^[73] 和矩阵分解算法^[74, 75] 是典型的基于模型的协同过滤算法。

不同于协同过滤算法，基于内容的推荐算法只需要用户和物品的描述文件就能够产生推荐结果，用户对物品的评分数据只作为一种附加信息。通过融合购买过的物品属性可以用来描述用户的偏好信息，以此形成用户的描述文件^[76]。算法通过依次比较候选列表中的商品，向目标用户推荐与用户描述文件最为相似的商品。基于内容的推荐算法主要使用关键字、标签和权重来描述商品，系统会为每个商品生成一份向量形式的摘要或抽象特征。目前，TF-IDF（term frequency-inverse document frequency）是使用最广泛的物品抽象特征提取技术。TF 表示单词词频，即单词在单个文档中出现的频率，数值越大代表单词越重要；IDF 表示逆文档频率，代表单词的普遍性，如果一个单词在很多文档中都出现过，那么这个单词就不能很好的用来区分文档。将 TF 和 IDF 的数值相乘作为关键字在向量中的权重，表示用于区分该文档和其他文档的贡献度。

近年来，基于网络结构的推荐算法在实际应用中取得了非常出色的成果，例

如资源分配算法^[77,78]和热传导算法^[79]。实验证明，基于资源分配的算法通常具有较好的推荐准确性，在资源的扩散过程中，由于流行商品（即大度物品）具有较多的连边，因此资源传递到该类物品的概率较高。这种现象使得整个商品集合中的冷门商品（即小度物品）得到的资源极少，被推荐的概率极低，不利于提高整个推荐系统的多样性和覆盖率。而基于热传导的算法与之正好相反，更多的资源被传递到小度物品中。Zhou 等人^[80]设计了一种结合资源分配和热传导算法的非线性混合模型，该模型在提高推荐准确性的同时还提高了推荐多样性，打破了传统思维中的多样性和准确率两难困境。Liu 等人^[81]利用参数控制将更多的资源流向大度用户以此增强了大度用户在热传导算法中的权重，使得热传导算法的推荐准确性得到了显著提升。在用户-物品二部图中，通过增加一些代表上下文信息的节点可以缓解系统数据稀疏的问题，同时还能更好的刻画用户之间和物品之间的相关性。Zhang 等人^[82]为用户-物品二部图添加了标签节点，将二部图扩展为用户-物品-标签三部图。Yao 等人^[83]根据用户和物品的属性为原二部图添加了不同类型的节点（例如用户的性别、物品的类型），形成了多层上下文网络结构（Multi-Layer Context Graph）。通常情况下，这些基于扩展网络的推荐算法能够提供更加准确的推荐结果。

1.2.3 时间感知推荐算法研究

与静态推荐算法的二维数据不同，时间感知的推荐算法是将时间信息作为上下文环境的动态推荐算法。定义时间集合 T ，兴趣函数 F 定义了如下的映射：

$$F : U \times O \times T \rightarrow R \quad (1-2)$$

时间维度对于推荐系统来说是一个十分重要的上下文信息，可以分别从用户和物品两个角度对其进行研究。对于用户来说，时间信息包括：年龄信息，它反映了用户在推荐系统中存在了多长时间；用户购买时间，即用户为物品打分的时刻；以及用户偏好模式，代表了用户随时间变化的打分模式。对于物品来说，时间信息包括：物品年龄信息，表示一个物品被用户评分后过去了多长时间；物品启动时间，即物品进入系统的时间；以及物品的流行度，其可以在较短时间内发生剧烈的变化以此反映物品受欢迎程度的时间特性^[84]。在时间感知的推荐算法中，时间信息分为与用户历史偏好相关的绝对时间（例如用户为物品打分的时间戳信息）和用户购买物品的相对时间（例如用户先后购买的物品序列）。

第一个考虑时间信息的推荐算法可以追溯到 2001 年。Zimdars 等人^[85]将协同过滤推荐问题视为单变量的时间序列预测问题，为此，他们对数据的时间相对顺

序进行了编码处理。使用时间信息最简单的方式是将时间信息作为上下文信息对数据进行预处理，剔除不大可能出现在推荐列表中的物品。例如：Tang 等人^[86]针对可扩展性问题，在电影推荐的上下文中考虑了电影发布的年份信息并利用它减少了候选电影的数量。Cho 等人^[76] 使用数据的购买序列来提升推荐结果的质量，其中购买序列同样包含着时间信息作为参数。Ding 等人^[87] 提出了一种基于新近性的协同过滤算法，该算法定义了一种基于概念漂移的新相似度度量方式。Nathanson 等人^[88] 开发了一个称为 Eigenstaste 的算法，该算法在选择下一个推荐物品时考虑了用户随时间变化的兴趣偏好，并按最近最多打分的方式重新排列了候选物品。随后，Lee 等人研究了若干种时序信息对协同过滤算法准确度方面的影响^[62]，并提出了一种将物品启动时间^[86]、评分时间^[87] 和隐性反馈相结合的伪评分数据，在伪评分数据上运行的推荐算法结果要比在原始数据上的更加精确。

Koren 等人提出的与时间相关的 timeSVD++ 方法^[39,74] 在赢得 Netflix 竞赛后，使得时间信息开始受到越来越多推荐系统领域研究人员的关注。在涉及多个物品和用户的推荐系统中，会同时存在多个局部特性发生漂移，而在传统概念漂移理论中通常仅跟踪一个特性。因此，Koren 强调推荐系统中的时间模型可能不同于传统的概念漂移。在他们提出的 timeSVD++ 算法中，预测评分可以通过如下公式得到：

$$\hat{r}_{ia} = \mu + b_a(t) + b_i(t) + q_\alpha^T \left(p_i(t) + |I(u_i)|^{\frac{1}{2}} \sum_{\beta \in R(u_i)} y_\beta \right) \quad (1-3)$$

其中， μ 表示评分的总体均值， $b_a(t)$ 是随时间改变的物品偏差， $b_i(t)$ 是随时间改变的用户偏差， q_α 和 y_β 是在联合隐因子空间 f 上的物品向量， $I(u_i)$ 表示被用户 u_i 打过分的物品集合， $p_i(t)$ 是在联合隐因子空间 f 上的用户向量，可以反映用户偏好随时间的变化。

在 Netflix Prize 竞赛中，Xiang 等人^[89] 也提出了一种时间相关的矩阵分解方法。他在模型的设计中考虑了多种随时间变化的因素，包括社会兴趣、用户偏好、物品受欢迎度和用户兴趣。此外，Xiang 在改进算法中还考虑了整体评分、用户行为和物品的季节周期性特性。Netflix Prize 竞赛提供的数据是按照用户个体随机划分的，即将每个用户的部分记录作为训练集，剩余的作为测试集。这种划分方式使得在预测某用户的评分时使用了其他用户的未来信息，这并不是推荐系统严格利用过去数据预测未来偏好的真实场景。因此，Lathia 等人^[90] 发现，一些可以在 Netflix 数据集中得到较高预测准确率的算法并不能在动态数据集中获得同样的效果。

以上讨论的所有时间感知的推荐算法都是关注在显性的评分数据上的。目前，

对二元数据进行的研究工作还很少。Xiang 等人^[91]通过基于网络结构的方法来捕捉用户在特定时间的偏好，研究了隐性反馈数据上时间因素的作用。他们提出了时间段图模型，在该图中用户的长期偏好通过用户与物品的连边表示，短期兴趣通过会话与物品的连边表示。算法通过在时间段图模型上的随机游走为目标用户产生推荐结果。Liu 等人^[40]提出了一种基于资源分配的动态推荐算法，该算法除了考虑数据的时间衰减外，还考虑了用户两次行为之间的时间间隔，认为相距越近的行为越是相似的。Song 等人^[92]研究了时间窗口对基于资源分配和热传导混合算法推荐性能的影响，发现只使用部分最近的数据记录依然能够得到较高的准确性。De Pesselier 和 Dooms^[93]也研究了时间信息对数据质量的影响，并进一步发现旧数据将有益于由内容提供商提供内容的系统，而不利于由用户制造内容的系统。

此外，Jambo 等人^[94]提出了一个鲁棒性很好的推荐算法，随着时间的演化，该系统利用现代控制理论的概念来提供最新的推荐结果。Ding 和 Li^[95]提出根据用户评分历史为每个用户使用个性化衰减因子，给旧数据分配更少的权重以反映时间在推荐算法中的重要性。基于最新产品是非常重要的而旧产品是微不足道的假设，Ren 等人^[96]提出了偏好模式的概念，用来表征用户偏好的时序动态。Huang 和 Huang^[97]提出了一种基于序列模式的推荐算法来预测用户随时间变化的购买行为。Chen 和 Han^[98]在基于内容的推荐算法中分析了序列模式对用户偏好的时间衰减作用。Lathia 等人^[99]认为常规准确度评价指标无法反映时间因素的作用，并提出了用来衡量推荐算法时序多样性的评价方法。

1.3 本文的主要研究内容与组织架构

本文各章节的关系如图1-4所示，分别从绝对时间和相对时间两个角度对用户的在线动态行为特性进行了深入研究。在研究绝对时间表现出的时间特性时，本文首先从用户的角度出发分别研究了显性反馈和隐性反馈的时间特性，然后从物品的角度对系统的时序特性进行了分析。通过观察到的现象，本文提出了多种时间感知的推荐算法以应对推荐系统的不同使用场景，以便提高推荐系统的推荐性能。具体的组织结构安排如下：

第一章讨论了课题研究的背景与意义，并从人类行为的动力学研究、静态推荐算法和时间感知的推荐算法这三个方面介绍了国内外的相关研究情况。

第二章首先介绍了推荐系统在商业领域的应用情况和面对的主要挑战，接着介绍了目前在学术领域关注最多的几种主要推荐算法，最后讨论了研究推荐算法需要用到的评价指标和公开数据集。

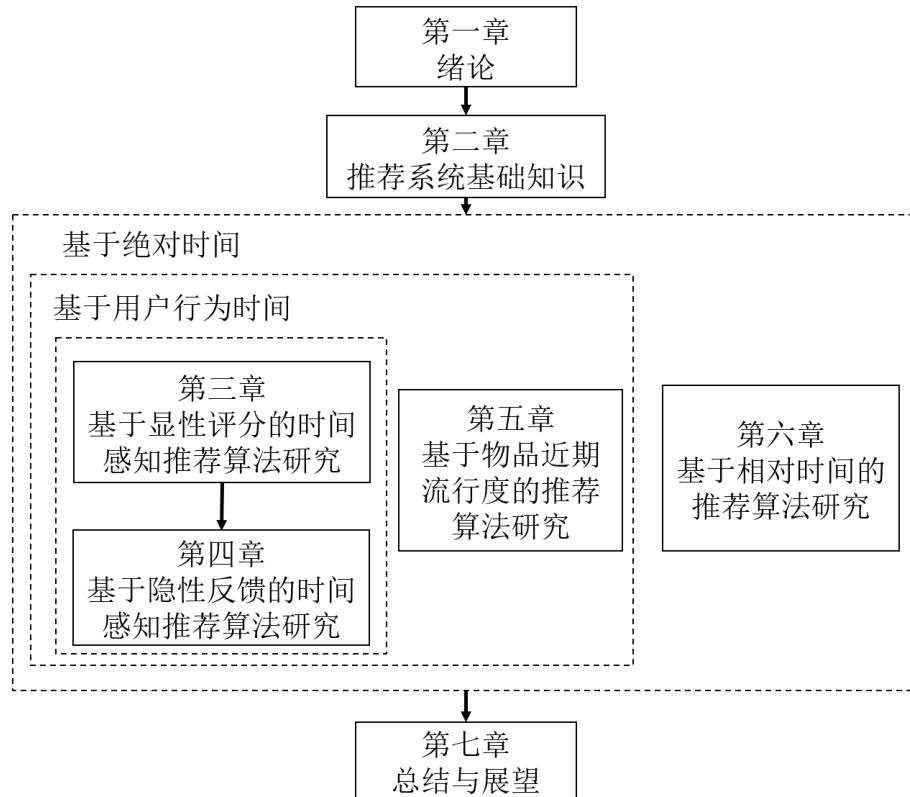


图 1-4 论文的章节关系

第三章主要研究了用户显性反馈行为中表现出的时间动力学特征。首先将用户对物品的显性评分分为正面和负面评价两种，分析了不同评价对用户未来兴趣的时序影响。然后提出了一种基于正负面评价的时间感知推荐算法，该算法同时考虑了用户的长短期偏好和正负面评价意见。为了提高算法的执行效率和在实际系统中的应用价值，利用线性回归方式对正负面评价的时间衰减曲线进行了近似拟合，通过数据来学习正负面评价的不同衰减因子。

第四章主要研究了用户在隐性反馈行为中表现出的时间动力学特征。首先分析了用户个体行为、用户整体行为和系统随机性行为的不同特性，然后根据用户行为的先后顺序，利用基于行为时间感知的划分方式，将系统中的用户分成了领导型用户和跟随型用户，并比较了不同类型用户的行为特性。最后提出了一种基于资源分配的时间感知推荐算法，并从准确性和个性化两个方面与其他基准算法进行了性能比较。

第五章研究了系统中物品近期流行度在推荐算法中的作用。首先从物品的角度分析了用户行为表现出的时间特性。然后研究了物品近期流行度，提出了基于物品近期流行度的资源分配算法和热传导算法。通过实验对比发现近期流行度能显著提高热传导算法的推荐性能。为了进一步研究近期流行度在热传导推荐算法

中的作用，将近期流行度融合到了三个基于热传导的改进算法中，并与原始算法的推荐性能进行了比较。最后，提出了一种基于近期流行度的后处理算法，将通过资源分配算法得到的物品评分进行了重新处理。

第六章主要研究了以相对时间表示的用户序列和物品序列在推荐算法中的应用。基于用户行为的相对顺序，提出了基于用户序列的资源分配算法和基于物品序列的资源分配算法。在实验过程中，首先研究了各参数对推荐算法性能的影响，然后通过不同算法的推荐结果比较，研究了用户序列和物品序列在推荐算法中的作用。

第七章是论文的总结和展望，在对目前工作进行简单总结的基础上，提出了下一步的研究任务和方向。

第二章 推荐系统基础知识

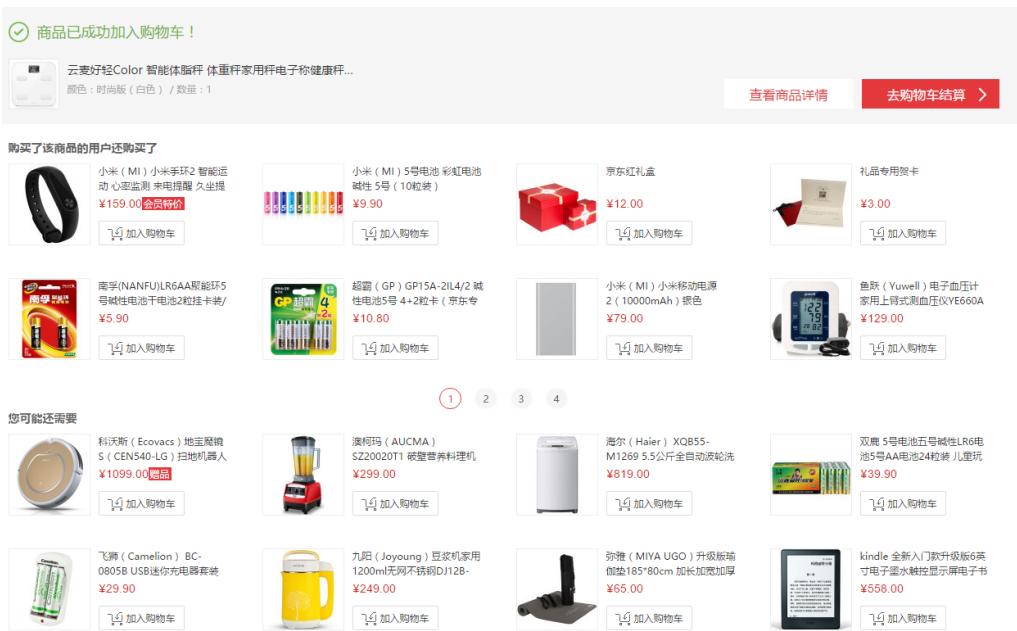
随着 O2O 行业的发展，众多企业将原有的线下模式转移到线上，形成了线上订单线下消费的经济模式，这大大扩展了推荐系统应用的场景。在大数据时代的背景下^[12,100]，应用研究已经成为当前推荐系统领域研究的重点。目前，已经有许多推荐算法被应用到不同的商业领域中，每种推荐算法都有各自的优点和局限性，例如协同过滤算法面临稀疏性、扩展性和冷启动问题^[63,64]，而基于内容的算法过于专业化^[63]。为了分析不同推荐算法在商业应用中的不足并加以改进，众多企业开始将自己的系统数据部分公开给研究人员，寄希望于通过群体智慧的方式促进推荐系统的长远发展。另一方面，为了解决推荐算法的线下评估问题，研究人员设计了多种离线评测指标，可以从准确性和个性化水平等不同的角度对算法进行评价。这些离线指标可以帮助企业减少测试环节的成本，快速比较不同推荐算法的优劣。为了更深入和全面的了解推荐系统，本章将介绍推荐系统在众多商业领域的应用情况、推荐系统面临的主要挑战、常用的推荐算法，以及研究推荐算法需要用到的评测指标和公开数据集。

2.1 推荐系统的商业应用

在向用户推荐商品以激发用户购买兴趣方面，推荐系统表现出很强的实用性，受到越来越多企业的关注。目前，推荐系统已经成功应用于多个行业的在线平台中，涵盖电子商务、电影和视频、数字音乐、新闻资讯和在线教育领域等。下面就推荐系统在这些热门商业领域的应用进行简要的介绍。

(1) 电子商务领域。电子商务是推荐技术应用最为成功的代表，在过去 20 年中得到了最多关注和研究^[88]。目前，各类电商网站都已经嵌入了推荐系统作为其核心业务功能，例如亚马逊、阿里巴巴和京东商城等。评分功能是电子购物网站最常用的辅助功能，客户能够通过对购买过的商品进行评分来表达自己的偏好。在没有提供评分功能的网站中，推荐系统也可以简单的将用户购买过的商品作为用户的兴趣，形成 0-1 的二元评分数据。评分数据通常被用于协同过滤推荐算法中，通过发现相似商品或者相似用户来为当前用户产生推荐结果。除了协同过滤算法，关联规则挖掘也是该领域的推荐系统常用的技术手段之一，通过分析用户的购物行为来发现强相关的商品（即经常被用户一起购买的商品）。商家可以将这些商品进行打包促销，增加商品的曝光度从而提高收益。但是关联规则挖掘一般无法提供像协同过滤算法那样个性化的推荐结果。此外，商品的属性信息被用

于基于内容的推荐算法中。系统通过商品间属性的相似度向用户推荐与挑选的商品相似度较高的其他商品。在电子商务领域，推荐系统最常见的使用场景是商品的详情页，系统根据物品的属性标签可以向用户展示同类商品的热销榜。图2-1是中国第二大在线零售平台京东商城的商品推荐界面。当用户将商品加入购物车后，推荐系统会为用户产生两类推荐结果。在图2-1下方的“您可能还需要”处展示的推荐商品是根据用户的购买行为采用协同过滤算法和基于内容的推荐算法产生的；而上方“购买了该商品的用户还购买了”处的商品主要通过协同过滤算法和关联规则将用户经常一起购买的强相关商品作为推荐商品，并且通过标题提示用户产生这些推荐商品的原因。

图 2-1 京东商城商品推荐界面^①

(2) 电影和视频领域。在电影和视频行业有许多成功应用推荐系统提高网站收益的成功案例。尤其是目前视频行业正在走向减少外部购买、加强自制节目和发展专业生产内容的道路上，越来越多的视频内容会不断出现在互联网上。为了将视频内容准确推送到目标人群中，众多视频网站开始使用推荐系统来分析网站用户、建立用户画像。类似于电子商务领域，协同过滤算法和基于内容的算法也是视频行业最常采用的推荐技术。在中国，优酷视频是历史最悠久的视频网站，并应用了很多推荐技术来提高在线用户的观影体验。当用户停留在某个视频页面时，网站会向用户展示三个方面的推荐内容，分别是“大家都在看”、“明星推荐”和“相关推荐”。在“大家都在看”模块中，系统使用基于物品的协同过滤算法向用

^① <http://www.jd.com>

户推荐观看过同一部视频的其他用户观看的视频内容；“明星推荐”模块根据当前视频的演员信息，向用户推荐剧中演员参演过的其他电影和电视剧，属于基于内容的推荐算法；“相关推荐”模块推荐的是网站用户的自制内容，可以采用关联规则或标签分析的方式产生对应的推荐结果。

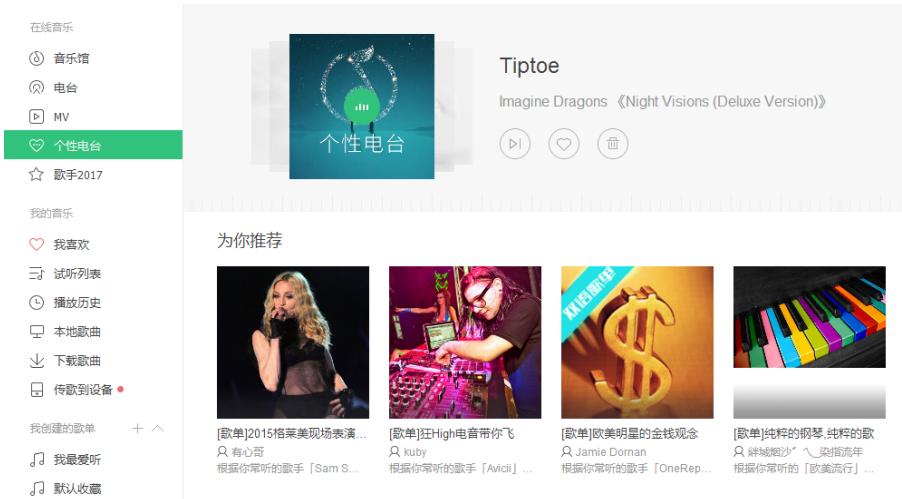


图 2-2 QQ 音乐客户端个性化电台推荐界面^①

(3) 数字音乐领域。在 20 世纪 80 年代，从一个刻有 10 首歌曲的胶带中选择一首喜欢的曲目是十分容易的。如今，随着移动智能手机技术的发展，16G 存储空间的手机可以存放超过 3000 首的音乐，用户想要知道所有歌曲的名字和歌手就变得十分困难了。在互联网中这种现象变得更加糟糕，网络在线音乐平台的兴起使得用户可以接触到市面上所有发布过的歌曲甚至是其他用户自己创作的曲目。推荐系统是数字音乐领域帮助用户挑选歌曲最合适的工具。该领域采用最多的推荐技术分别是专家系统和协同过滤算法。例如，著名的个性化网络音乐电台 Pandora 构建了一个音乐基因系统，为每首歌曲根据不同的元素（例如旋律、节奏和歌词等）进行了标注，形成歌曲的属性特征，系统根据这些属性特征采用基于内容的推荐算法来为用户产生推荐结果。Last.fm 通过采集用户对歌曲的反馈信息计算用户与用户之间的相似度，并在此基础上向用户推荐相似用户收听过的歌曲。Chen 等人^[101] 通过让用户在收听歌曲前主动输入想要的歌曲信息，将信息检索技术融入到了音乐推荐系统中。图2-2是国内知名音乐平台 QQ 音乐的个性电台界面。QQ 音乐根据用户的历史收听记录采用协同过滤等算法为用户产生了推荐歌曲列表，并将推荐的歌曲组成个性电台。在个性电台中，用户不能自己挑选歌曲，只能通过下一首、喜欢和不喜欢提供隐性反馈。系统根据用这些隐性反馈对个性化推荐

^① <https://y.qq.com/>

结果进行修正。

(4) 新闻资讯领域。随着移动智能终端的普及，用户接触媒体的方式呈现出移动化、碎片化、多元化、互动性的趋势。移动端（手机和平板电脑）已经成为最重要的新闻资讯获取途径，资讯的接收方式呈现出更加便捷、快速和定向的特点。在互联网上，信息很容易在短时间内爆发式的传播，谣言也更容易扩散。与资讯内容的新颖性相比，用户更加看中资讯内容的可靠性，用户更愿意从专业媒体和官方媒体获取资讯信息。因此，在如今快节奏的城市化生活中，使用推荐系统分析用户的阅读兴趣，在用户的碎片化时间内向其推荐相关资讯变得越来越有价值。在大多数情况下，新闻资讯的文本内容都可以提取成对内容高度精炼的关键词列表。许多推荐系统针对该领域的设计都是建立在对关键词的分析工作上，概率模型是这一领域最常用的技术。在新闻推荐系统中，上下文资源被转换为向量形式，其中每个元素可以表示为考虑了频率和位置信息（例如是标题或纯文本）的关键词。系统通过检索与用户模式相似的新闻文本来产生推荐。如果用户能够对新闻进行评分，那么协同过滤算法也是可行的方法之一。例如，新闻推荐系统 News Duce^[102] 使用协同过滤算法来为用户短期兴趣进行建模。图2-3是著名新闻聚合平台今日头条的网站首页，用户每次打开或刷新首页时，系统会根据当前的热点实事和用户过去的阅读兴趣为用户推送 6 条推荐文章。除了首页以外，另一个新闻资讯推荐场景是新闻详情页，系统会推荐与用户正在阅读的内容高度相似的其他文章。



图 2-3 今日头条首页界面^①

(5) 在线教育领域。随着互联网的快速发展，以公开课和在线视频为代表的在线教育受到越来越多年轻人的欢迎，他们可以利用移动智能手机在平时碎片化的

^① <http://www.toutiao.com/>

时间内学习以前无法接触到的知识。在线教育领域的推荐系统旨在帮助学习者选择他们感兴趣的课程、学习材料，以及学习活动（例如课堂讲座或在线学习小组讨论）。在线教育领域的资源内容往往是视频这类多媒体资源，同时用户对课程也很少提供显性的评分数据。这就造成了系统往往缺乏足够的数据来支撑协同过滤算法或者基于内容的推荐算法，因此基于知识的教育规则在在线教育推荐系统中扮演着更加重要的角色。Zaiane^[103] 构建了一个软件代理，该软件代理使用诸如关联规则之类的数据挖掘技术来构造用户在线行为的模型，并且使用该模型来推荐活动或快捷方式。学习者可以通过使用推荐的快捷方式来快速的发现相关资源。Lu^[104] 开发了一个个性化电子学习资料推荐系统（PLRS）。当创建了学习材料数据库或学习活动数据库并且系统获得学习者的注册信息后，PLRS 使用计算分析模型来识别个人的学习需求，然后通过规则匹配来为学习者生成学习资料（或活动）的推荐。

2.2 推荐系统面临的主要挑战

虽然推荐系统已经成功应用于众多商业领域，帮助用户和企业建立起了沟通的桥梁，但是随着互联网的发展，用户的使用场景不断的发生改变，新的需求和新的问题层出不穷，推荐系统依然面临着众多的挑战。只有深入了解这些挑战和问题，才能设计合适的推荐系统以应对不同的需求。

(1) **冷启动问题**。因为新用户的个人描述文件基本是空缺的，同时罕有对物品的打分和浏览信息，导致系统对新用户的偏好口味了解甚少，无法给出精确的推荐。类似的，系统中会出现很多新的商品，根据长尾理论的概念其中有大量的商品仅被很少的用户购买过，这些新的或者冷门的商品也很难找到合适的目标用户进行推荐。这两类现象被称为冷启动问题。在文本类的商品销售系统中，冷启动问题会相对比较少。因为在这类系统中，物品的内容可以通过文本信息进行提取，从而计算物品间的相似度。而在多媒体内容提供平台中（例如音乐、视频等），冷启动问题就比较突出了。

要解决或缓解冷启动问题，核心思路是尽可能多的收集用户的需求信息。在一些推荐系统中，用户注册成功后会被要求填写个人信息，例如年龄、居住城市、受教育程度、性别和职业等，系统以此构建用户的个人模型^[105,106]。另外，用户为物品打的标签信息也可以用来缓解冷启动问题。标签是用户为物品提取的精炼信息，能够从侧面反应用户的偏好和物品的内容^[82]。此外，新用户更容易选择特别流行的商品，使用热销榜能在一定程度上为新用户推荐合适的商品。

(2) **数据稀疏性问题**。稀疏性是指系统中有用信息的缺失问题。如今，互联网

系统的规模变得越来越庞大，系统中不仅有数以万计的用户，更有数以千万计的商品。如果用户和物品的交互行为很少，即用户只选择了很少的一部分物品，系统就会产生数据稀疏的问题。一方面，用户只对很少的物品进行过选择，无法准确的刻画用户的口味偏好；另一方面，用户之间选择的物品重叠度也非常低，很难准确的比较用户之间偏好的相似度。

稀疏性问题是任何系统都面临的问题，无法从根本上完全解决。为了缓解这个问题，研究人员提出了很多方法。最简单的方法是将物品信息粗粒度化，不再关注用户对具体物品的打分信息，而是考虑用户购买过的物品的类别信息，通过计算类别之间的相似度，构建基于类别的推荐算法。另外，基于资源分配的算法可以考虑系统的全局信息关联性，从原来的一阶关联（两个用户购买的共同物品）扩展到二阶或更高阶的关联性^[107-109]。研究人员还通过在用户-物品评分矩阵中添加缺省的打分来提高打分矩阵的稠密度，以此提高了推荐准确率^[110]。

(3) 扩展性问题。除了系统本身已经拥有的大批用户和商品外，新的用户和新的商品也在源源不断的加入到系统中，用户与商品间发生的交互信息也越来越多。如何快速有效的处理数据已经成为当前各类系统急需解决的问题。尤其是在推荐系统领域，对推荐算法的空间和时间复杂度的要求越来越高，推荐系统需要根据用户的最新偏好尽快为其推送最新的物品。

除了设计更为简单的模型应用于大规模变化的系统外，并行化是解决扩展性问题的又一有效途径。例如，Google 中国在 Orkut 系统中开发的 LDA 并行化算法^[111] 以及 Gemulla 等人提出的随机梯度下降法^[112] 都是解决推荐系统扩展性的有效手段。另一种应对增量数据的方法是开发增量计算模型，增量计算模型旨在通过考虑新增节点和新增连边的情况下，对已有结果进行微调进而得到新的结果^[113]。通常情况下，随着数据量的不断增加，增量计算模型需要在一定时间后重新利用全局信息计算一次结果，以保证数据结果的同步性。

(4) 准确性和多样性的两难问题。口碑传播在商品销售过程中起着十分重要的作用，用户的选择行为会受到来自亲朋好友意见的影响，尤其是在高度成熟的社交网络中。给用户推荐热门产品通常可以得到很高的推荐准确率，但通过这种方式推荐的商品往往是用户已经从其他途径知道的商品，用户从中几乎获得不到有用的价值。只对用户推荐热门商品会造成明显的马太效应，热门的商品越来越流行，冷门的商品却一直无人问津。如何准确的向用户推荐感兴趣的冷门商品被看作是推荐系统在商业领域的核心价值。一方面，这会使得商家可以卖出更多的长尾商品，获得更多的收益；另一方面，用户可以获得意想不到的惊喜，增强用户的满意度和用户粘性。

提高多样性和提高准确性一般被认为是矛盾的，提高多样性会使得用户获得的推荐列表中出现很多冷门物品，一旦用户不喜欢的物品出现过多，会使得用户对推荐系统的可靠性产生严重的怀疑。一种提高多样性的有效方式是后处理技术，在得到推荐结果以后对结果再次进行处理，从而提高多样性^[114, 115]。另一种方式是在推荐模型中综合考虑准确性和多样性，例如结合资源分配和热传导的混合算法^[80]。

(5) **脆弱性问题**。目前各行各业的在线网站中都应用了推荐系统，以刺激更多的用户产生消费行为，增加企业的收入。在作为第三方平台的电商系统中，由于商家之间存在明显的竞争关系，如何让自己的商品脱颖而出是商家最关心的事情。为了让自己的商品得到更多的曝光度，一些商家想出了各种恶意的手段来提高自己商品被推荐的概率^[116]，例如恶意评分、刷榜等行为。因此推荐系统需要提供很好的鲁棒性以应对恶意攻击，保证推荐结果的公正和有效。常用的技术手段主要是通过分析真实用户和恶意用户之间的购物行为差异，主动的发现恶意行为从而从数据输入方面提高系统的鲁棒性^[117-119]。

(6) **系统评估问题**。推荐系统已经发展了几十年，在这过程中提出了很多用于评价推荐系统性能的指标。从应用场景的观点出发，可以将评价指标分为线上评价指标和离线评价指标。离线指标包括准确率、召回率、多样性、覆盖率等，主要适用于在线下小数据上对算法进行评估，评估内容更多关注于数据本身。需要注意的是，离线指标存在结果不一致的问题，例如评分预测指标的提高不一定能提高推荐列表的排序准确率。线上指标包括转化率、购买率、客单价等，是企业更加关注的行业指标，评估内容更多关注于用户。离线指标和线上指标的打通可以极大的帮助企业减少部署和测试推荐算法的成本，但是到目前为止，他们之间的关联性依然缺少深入的研究和理论证明。

(7) **用户隐私问题**。隐私问题是推荐系统需要解决的又一个问题。为了得到准确的推荐结果，在线推荐系统通常需要尽可能多的收集用户各种信息，包括人口统计学信息、历史行为信息、地理位置信息等。如果用户发现系统在为其产生推荐时，暴露了自己的隐私，会使得用户缺少安全感，从而不愿意向系统提供更多的信息甚至可能产生用户流失。例如，京东商城曾被披露有数千万用户的个人信息遭到泄露，这引起了大批用户的担忧，从而造成部分客源的流失。随着智能手机的普及，用户随时随地都在留下自己的足迹，相关研究表明通过组合不同来源的用户信息已经可以推断出用户的许多个人隐私^[120]。因此，如何在提供云端推荐引擎服务的第三方平台（例如阿里云的推荐引擎服务）中保护用户隐私就显得十分重要。为了保护用户在系统中存储的个人信息，诸如混合方法^[121] 和模糊

处理^[122]等技术被加入到了推荐系统中。

2.3 常用的推荐算法

推荐系统的目的是向用户推荐他们最有可能喜欢和接受的产品，包括电子服务和实物。自从推荐系统成为学术界和工业界的热门研究领域以来，已经有许多不同类型的推荐算法被提出。常用的推荐算法包括协同过滤推荐算法、基于内容的推荐算法、基于上下文的推荐算法，基于网络结构的推荐算法和混合推荐算法等。通常来说，协同过滤算法是基于评分矩阵来进行预测的，而基于内容的算法在生成推荐结果时使用的是商品内容。基于上下文的算法在计算过程中采用了额外的附加信息。基于网络结构的推荐算法将推荐问题视为在二部图网络结构中的连边预测问题。本节就这些常用的推荐算法进行简单介绍。

2.3.1 协同过滤推荐算法

当尝试向目标用户 u_i 推荐商品时，协同过滤算法首先通过查找具有相似评分模式的其他用户，然后推荐这些相似用户评分较高的商品。协同过滤算法的基本假设是：如果两个用户 u_i 和 u_j 对一些商品的评分比较相似，那么他们对另外一些商品的评分或消费行为也将是相似的。在这个假设条件下，协同过滤算法主要或者有时仅基于评分矩阵来产生推荐结果。协同过滤算法可以分为基于内存的方法和基于模型的方法^[123]。

2.3.1.1 基于内存的协同过滤算法

Resnick 等人^[18]于 1994 年提出的基于内存的协同过滤算是该领域的开创性工作。他们提出的 GroupLens 系统利用评分矩阵来识别与目标用户具有相似行为的用户，并且能够预测目标用户对特定商品的评分。协同过滤算法涉及两个阶段：邻居选择和评分聚合。邻居选择指通过评估用户间的相似度来为目标用户 u_i 确定 k 个与之最相似的用户组成邻居集合 $N_k(u_i)$ ；评分聚合指将邻居集合中的用户对特定物品 o_α 的评分数据进行聚合作为目标用户 u_i 对物品 o_α 的预测评分。

在邻居选择阶段，用户 u_i 的邻居集合 $N_k(u_i)$ 是根据评估用户间相似度确定的。两个最流行的相似度度量方法分别是 Pearson 相关性系数（Pearson correlation coefficient，简称 PCC）^[94] 和余弦相似度（Cosine similarity，简称 COS）^[110]，公式定义如下：

$$sim(u_i, u_j)_{PCC} = \frac{\sum_{o_\alpha \in O_{ij}} (r_{ia} - \bar{r}_i)(r_{ja} - \bar{r}_j)}{\sqrt{\sum_{o_\alpha \in O_{ij}} (r_{ia} - \bar{r}_i)^2 \sum_{o_\alpha \in O_{ij}} (r_{ja} - \bar{r}_j)^2}} \quad (2-1)$$

$$\text{sim}(u_i, u_j) \cos = \cos(\mathbf{u}_i, \mathbf{u}_j) = \frac{\sum_{o_\alpha \in O_{ij}} r_{ia} r_{ja}}{\sqrt{\sum_{o_\alpha \in O_{ij}} r_{ia}^2} \sqrt{\sum_{o_\alpha \in O_{ij}} r_{ja}^2}} \quad (2-2)$$

其中, O_{ij} 表示用户 u_i 和 u_j 共同打过分的商品集合, \bar{r}_i 和 \bar{r}_j 分别表示用户 u_i 和 u_j 的平均打分值。

在评分聚合阶段, 针对任意物品 o_α , $N_k(u_i)$ 集合中所有用户对于物品 o_α 的评分将通过以下三种方法聚合在一起作为用户 u_i 对该物品的预测评分值 \hat{r}_{ia} ^[124]:

$$\hat{r}_{ia} = \begin{cases} \frac{1}{k} \sum_{u_j \in N_k(u_i)} r_{ja} & (a) \\ \eta \sum_{u_j \in N_k(u_i)} \text{sim}(u_i, u_j) \times r_{ja} & (b) \\ \bar{r}_i + \eta \sum_{u_j \in N_k(u_i)} \text{sim}(u_i, u_j) \times (r_{ja} - \bar{r}_j) & (c) \end{cases} \quad (2-3)$$

其中, $\eta = 1 / \sum_{u_j \in N_k(u_i)} \text{sim}(u_i, u_j)$ 作为一个归一化因子, \bar{r}_j 表示用户 u_j 的平均打分值。公式2-3(a) 仅仅简单的将邻居用户的评分均值作为预测分数, 而更为流行的方法是公式2-3(b) 中所示的通过加权融合的方式, 用户之间的相似度被作为衡量邻居用户对目标用户的重要性。公式2-3(c) 克服了不同用户的评价尺度不一致的缺点, 考虑了不同用户平均打分的偏差。

2.3.1.2 基于模型的协同过滤算法

基于模型的协同过滤算法是通过给定的评分矩阵来直接构建模型, 并利用该模型来预测用户对物品的喜爱程度。一系列机器学习技术被应用于基于模型的协同过滤算法中, 例如监督学习、无监督学习和矩阵分解技术。

(1) **基于监督学习的协同过滤算法**。对于二元的评分矩阵, Billsus 和 Pazzani^[125] 将推荐问题看作一个二分类问题, 该问题可以通过使用诸如神经网络的方法解决。对于一般的数值评分矩阵, Su 和 Khoshgoftaar^[126] 利用信念网络将其视为多分类问题。此外, 贝叶斯分类器可以用于将基于用户和基于物品的协同过滤算法进行融合, 实验结果表明融合算法的推荐结果要优于单独使用的结果^[127]。

(2) **基于非监督学习的协同过滤算法**。基于聚类的协同过滤方法通过将相似的用户分组在一起, 并使用群组作为邻居节点来提高了算法的扩展性^[128], 但是不一定能产生更准确的结果。双聚类技术可以同时对用户和物品进行分组, 这可以进一步降低协同过滤算法的计算成本^[129]。此外, Lemire 和 Maclachlan^[130] 考虑到用户的评分行为同时受用户的评分风格(例如用户的评分均值)和物品的流行度(例如物品的打分均值)影响, 由此发明了 Slope one 推荐算法。

(3) 基于矩阵分解的协同过滤算法。Billsus 和 Pazzani^[125] 提出利用 SVD 技术来获取用户评分数据中的潜在信息，该方法可以利用与目标用户原本没有关联性的用户的评分信息为目标用户产生推荐结果。在这之后，研究人员提出了许多基于 SVD 的推荐算法，例如融合了 SVD 和基于邻域算法的 SVD++ 算法，该方法同时使用了用户的显性反馈和隐性反馈信息^[131]。此外，Ren 等人^[132, 133] 通过利用评分矩阵中的全局评分模式，以迭代的方式将每位用户投影到低维子空间中，解决了数据稀疏性问题。

2.3.2 基于内容的推荐算法

当试图向目标用户 u_i 推荐商品时，基于内容的方法将通过商品的各种特征来为其构建分类器，然后在不考虑其他用户模型的情况下为该用户提供推荐结果。具体来讲，算法将每个候选商品 o_α 的内容与目标用户的偏好描述文件进行比较，向其推荐具有最高共同度的商品。

基于内容的推荐算法的第一个问题是表示用户。定义 \vec{v}_α 表示物品 o_α 的内容， \vec{f}_i 表示目标用户 u_i 的偏好描述文件。对于基于文本的商品， \vec{v}_α 可以使用诸如 TF-IDF 的方法来表征，将商品表示为基于内容提炼的关键词向量。将物品内容看作是一篇文档，从中抽取出若干关键词，关键词 t 在文档 d 中的 TF 值等于：

$$TF_{t,d} = \frac{N_{t,d}}{\sum_k N_{k,d}} \quad (2-4)$$

其中， $N_{t,d}$ 表示关键词 t 在文档 d 中出现的次数。关键词对应的 TF 值越大，代表该词越能代表当前文档。关键词 t 的 IDF 值可以通过总文档数与包含关键词 t 的文档数的比值取对数得到，计算公式如下：

$$IDF_t = \log \frac{|D|}{1 + |d \in D : t \in d|} \quad (2-5)$$

其中， D 表示文档集合，分母表示包含关键词 t 的文档数，分母加 1 是避免某关键词是从其他语料库中选取，但在本语料库中未出现时导致分母为 0 的情况。物品 o_α 可以表示为向量 $\{w_{1,\alpha}, w_{2,\alpha}, \dots, w_{k,\alpha}\}$ ，其中

$$w_{t,\alpha} = TF_{t,\alpha} \times IDF_t \quad (2-6)$$

基于商品内容的向量表示方式，通过将用户 u_i 购买过的所有物品的内容向量聚在一起可以得到用户的偏好描述文件 \vec{f}_i 。

另一个问题是如何衡量物品与目标用户的共同度。共同度可以表示为物品与目标用户向量形式下的相似度，余弦相似度是最常用的方法：

$$\text{sim}(o_\alpha, u_i) = \frac{\sum_{k=1}^N v_{\alpha k} p_{ik}}{\sqrt{\sum_{k=1}^N v_{\alpha k}^2 \sum_{k=1}^N p_{ik}^2}} \quad (2-7)$$

其中， N 是物品内容向量的维度， $v_{\alpha k}$ 和 p_{ik} 分别表示 \vec{v}_α 和 \vec{f}_i 的第 k 个元素。除了启发式方法，机器学习方法也可以用来计算物品与用户的共同度。

2.3.3 基于上下文的推荐算法

除了物品的内容、用户偏好描述文件和历史评分信息以外，上下文信息也对消费行为有显著的影响。例如，对于电影推荐场景，何时、何地以及和谁一起观看电影将对用户选择观看的电影产生明显的影响。研究人员已经认识到上下文信息可以进一步用于改善推荐质量。利用可用的上下文信息，推荐过程可以视作在 n 维空间上的函数 F 的映射，而不是在传统的二维空间中：

$$F : D_1 \times D_2 \times \cdots \times D_n \rightarrow R \quad (2-8)$$

Baltrunas 和 Ricci 的工作^[134] 考虑了一种特殊形式的上下文维度——时间信息，他们根据时间相关的因素（例如用户打分时的季节因素）将物品评分记录划分到了不同的子集中。其他上下文因素也能够通过该方法融入到现有推荐模型中。Adomavicius 和 Tuzhilin^[23] 将这种推荐系统定义为上下文感知的推荐系统，并且提出了两种算法范例来将上下文信息融合到推荐过程中：

(1) 预处理 (Prefiltering, 简称 PreF): 上下文信息用于在运行常规推荐算法之前预先处理数据。例如，首先根据特定的上下文环境预处理候选项，去掉不满足基本条件的商品。这种方法既能提高计算效率，又能缓解数据稀疏性问题。

(2) 后处理 (Postfiltering, 简称 PoF): 在应用常规推荐算法后，利用上下文信息对推荐结果进行重新处理。例如，推荐结果可以根据与特定上下文的相关性进行调整，剔除与特定上下文信息几乎没有相关性的商品。该方法还能与其他算法构成串联的推荐模型，将自身作为其他算法的预处理模块。

2.3.4 基于网络结构的推荐算法

从网络结构分析的角度看，推荐系统可以视为一个二部图网络上的链路预测问题，其中连边仅允许连接来自不同集合的节点：用户集合 U 和物品集合 O 。当用户 u_i 购买过物品 o_α 时，则存在连接节点 u_i 和节点 o_α 的连边。基于网络结构的

推荐算法的一个基本假设是：用户 u_i 对物品 o_α 进行评分可以表示用户 u_i 对物品 o_α 是感兴趣的，并且这种兴趣可以通过连边在网络中进行扩散。

受基于网络结构的资源分配动力学技术的启发，Zhou 等人^[77] 提出了一种基于网络结构的推断方法，该方法以基于资源分配的随机游走过程生成推荐结果。对于目标用户 u_i ，可以将他购买过的物品的初始偏好设置为：

$$AC(o_\alpha) = \begin{cases} 1 & \text{如果 } o_\alpha \in I(u_i) \\ 0 & \text{否则} \end{cases} \quad (2-9)$$

其中， $I(u_i)$ 表示用户 u_i 已经购买过的物品集合。如果将物品的偏好作为其分配的资源，这种资源可以通过以下两个阶段在所有物品间进行重新分配，最终具有最高资源的物品将会推荐给目标用户 u_i 。

传播阶段 (Spreading step): 在这个阶段，所有资源将从物品节点流到用户节点，分配给用户 u_j 的资源可以通过以下公式计算得到：

$$AC(u_j) = \sum_{\alpha=1}^n \frac{a_{j\alpha} AC(o_\alpha)}{k_\alpha} \quad (2-10)$$

其中， k_α 表示物品 o_α 的度（即有多少用户购买了物品 o_α ）， $AC(u_j)$ 是用户节点 u_j 上获得的资源数， $AC(o_\alpha)$ 表示由用户 u_i 分配给物品 o_α 的初始资源数，如果用户 u_j 与物品 o_α 之间存在连边则 $a_{j\alpha} = 1$ ，否则 $a_{j\alpha} = 0$ 。

重新分配阶段 (Redistribution step): 在这个阶段，资源将从用户节点重新流向物品节点，分配给物品 o_α 的最终资源 $AC'(o_\alpha)$ 可以通过以下公式计算得到：

$$AC'(o_\alpha) = \sum_{j=1}^m \frac{a_{j\alpha} AC(u_j)}{k_j} \quad (2-11)$$

结合公式 (2-10)，公式 (2-11) 可以重新写成：

$$AC'(o_\alpha) = \sum_{\beta=1}^n w_{\alpha\beta} AC(o_\beta), \text{ 其中 } w_{\alpha\beta} = \frac{1}{k_\beta} \sum_{j=1}^m \frac{a_{j\alpha} a_{j\beta}}{k_j} \quad (2-12)$$

图2-4给出了一个基于网络结构的推荐算法示例，其中三角代表用户节点，方块代表物品节点，黑色实心三角代表目标用户。首先，如图2-4(a)所示，将目标用户购买过的物品的初始资源设置为1，未购买的设置为0；然后资源从物品节点按物品的度分之一的概率传递到邻居用户节点上，如图2-4(b)所示；最后如图2-4(c)所示，将用户节点得到的资源按用户度分之一的概率再次传递回物品节点中。最

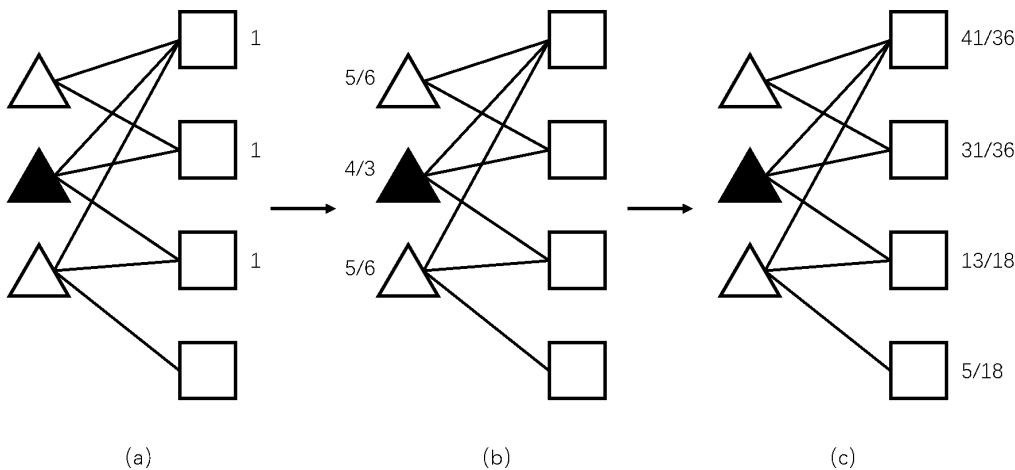


图 2-4 二部图上资源分配算法示意图

终，每个物品节点得到的资源即为目标用户喜欢该物品的可能性，系统通过剔除用户已经购买过的物品，将剩余的物品按获得资源的大小降序排列，资源最大的前 L 个物品推荐给用户。

该过程中一个局限是分配给物品的初始资源与流行度成正比，这放大了流行物品的作用^[135]。此外，虽然不同的用户购买了相同物品，但是他们可能是对物品不同的属性特征感兴趣。在传播的过程中，物品的相同属性的作用会被累加起来，这造成了冗余问题。文献[108]通过采用二阶权重矩阵来缓解该问题。当系统中含有标签信息时，二部图网络可以扩展成三部图网络，该图是用户-物品网络和物品-标签网络的融合。Zhang等人^[82]提出在标签信息可用时，可以利用三部图网络来缓解冷启动问题。

2.3.5 混合推荐算法

为了克服不同推荐算法的局限，可以在推荐过程的不同阶段将各种算法进行混合，发挥不同类型算法的优势，提高推荐系统的性能：

(1) 信息提取阶段：这种方式尝试将不同算法使用的各种输入信息进行整合。例如像 Fab 算法^[136]，当评分矩阵比较稀疏时，或者两个用户间没有共同购买的物品时，物品的内容为缓解协同过滤算法冷启动问题提供了有价值的信息。Ling 等人^[137]利用概率生成模型将用户的评分数据和评论信息进行了融合，在提高推荐准确率的同时缓解了冷启动问题。此外，社交信任关系已经被成功的应用于协同过滤算法中^[138-140]。

(2) 推荐生成阶段：此类方法尝试将不同的推荐算法聚合到统一的模型中。一个最常见的策略是对数据使用各种推荐算法分别生成独立的推荐结果，然后通过

投票的方式产生最终的推荐列表^[141]。另一种方式是通过应用一些数学模型直接将不同的算法进行组合。Jahrer 等人^[142]对不同的集成方式进行了研究，结果表明当与神经网络、梯度集成决策树和基于核函数的岭回归等方法进行比较时，线性回归方式是一种可接受的次优解。

2.4 推荐系统评测指标

推荐系统的评测方式可以分为在线评测和离线评测。在线评测是指用户直接通过系统对推荐结果给予反馈，或者参加调查问卷对结果进行评价。这种方式可以得到较为准确的评测结果，但是评测的成本一般很高，用户的参与度也不高。目前使用更多的是利用线上 A/B 测试为不同的用户部署不同的算法，最后依据不同算法得到的转化率等商业指标来衡量算法的优劣。由于在线部署和评测新算法成本昂贵，或者研究机构不具备真实的在线系统，离线评测成为了验证算法有效性的最主要手段。离线评测中研究人员最关心两个方面的指标，分别是准确性指标和个性化指标。准确性指标用来衡量推荐结果的精度，越高的准确性表示对用户兴趣的预测越正确。个性化指标用来衡量推荐结果的差异性，评判推荐系统能否给不同的用户产生不一样的推荐列表，能否把小众商品尽可能多的展示给用户等。

2.4.1 准确性指标

(1) 准确率。对于用户 u_i ，准确率^[143-145]表示系统为其提供的长度等于 L 的推荐列表中真实喜欢的物品（即出现在测试集中的商品）所占的比例，用户 u_i 的推荐准确率 $P_i(L)$ 定义为：

$$P_i(L) = \frac{h_i}{L} \quad (2-13)$$

其中， h_i 表示用户 u_i 的推荐列表与测试集中重叠的物品数量。例如，向用户推荐的 10 个商品中有 8 个是他真实喜欢的，那么系统为该用户推荐的准确率等于 80%。所有用户的平均准确率即为系统的推荐准确率：

$$P(L) = \frac{1}{|U|} \sum_{i=1}^{|U|} P_i(L) \quad (2-14)$$

其中， U 表示所有用户的集合。

(2) 召回率。召回率^[143-145]指推荐列表中真实喜欢的物品占该用户测试集中全部选择的物品的比例，用来衡量推荐系统的查全率。用户 u_i 的召回率 $R_i(L)$ 定义为：

$$R_i(L) = \frac{h_i}{n_{ip}} \quad (2-15)$$

其中, n_{ip} 表示测试集中用户 u_i 选择的物品数量。例如, 向用户推荐的 10 个商品中有 8 个是他真实喜欢的, 而测试集中该用户选择过 50 个商品, 那么对于该用户的推荐结果召回率就等于 16%。所有用户的平均召回率即为系统整体的召回率:

$$R(L) = \frac{1}{|U|} \sum_{i=1}^{|U|} PR_i(L) \quad (2-16)$$

(3) **F 指标。**准确率和召回率都是依赖于推荐列表长度 L 的准确性评价指标, 不同长度的推荐列表可以得到不同的准确率和召回率。通常情况下, 准确率和召回率中任意一个指标的增加会导致另一个指标的降低, 形成一个类似跷跷板的关系。例如在极端情况下, 推荐系统为用户只推荐一个结果并且是正确的, 那么准确率就会是 100%, 而此时召回率就会很低; 而如果把所有该用户选择过的物品都推荐给他, 那么召回率就会是 100%, 但是准确率此时往往很低。为了将准确率和召回率统一在一起考虑, 研究人员提出了 F 指标^[143-145], 计算公式如下:

$$F(L) = \frac{(\alpha^2 + 1)P(L) * R(L)}{\alpha^2(P(L) + R(L))} \quad (2-17)$$

其中, α 表示准确率和召回率的比重关系。在大部分研究中, 都将准确率和召回率给予相同的权重, 即将参数 α 设置为 1, 此时的 F 指标称为 F1 指标:

$$F1(L) = \frac{2 * P(L) * R(L)}{P(L) + R(L)} \quad (2-18)$$

(4) **排序准确率。**在对推荐排序很敏感的系统中, 用准确率指标来衡量系统性能变得不再合适。例如有两个推荐算法分别为用户推荐的 10 个商品中都正好有 1 个是用户喜欢的, 则它们的推荐准确率都为 10%。但是如果第一个算法将该物品排在推荐列表首位, 而另一个算法将该物品排列在第三位, 则在该种场景下, 第一个算法应该被认为是性能更优的。排序准确度越小表示系统将用户喜欢的物品尽可能的排在了推荐列表前端, 反之则表明喜欢的物品被排在了列表尾部, 越小的排序准确度代表越好的推荐性能。文献 [135] 提出了排序准确率指标来度量推荐列表的排序精度, 其计算公式如下:

$$RS_i = \frac{1}{|T_i|} \sum_{\alpha \in T_i} \frac{L_i^\alpha}{N_i} \quad (2-19)$$

其中, L_i^α 表示用户 u_i 在测试集中选择的物品 o_α 在推荐列表中排列的位置, T_i 表示用户 u_i 测试集中的物品集合, N_i 表示用户 u_i 未购买过的物品数量。所有用户的平

均排序准确率即为系统整体的排序准确率：

$$RS = \frac{1}{|U|} \sum_{i=1}^{|U|} RS_i \quad (2-20)$$

(5) **AUC 指标**。由于真实系统中大多数数据存在类别不平衡的现象，即负样本比正样本多很多的情况，同时，测试数据集中的正负样本的分布也可能随时间发生变化。这些情况会使得有些评价指标会发生明显变化，例如 Precision-Recall 曲线。ROC 曲线^[143]可以很好的在分类问题中解决这类问题。但是 ROC 曲线的绘制较为繁琐，且无法定量的衡量算法效果，实际研究中通常采用 AUC 指标^[145]，即 ROC 曲线下的面积，来定量的区分哪一个算法的性能更优。AUC 指标可以采用如下方法近似计算：针对一个目标用户，每次从测试集购买过的商品中随机选择一个商品 o_α ，然后从全部未购买过的物品中随机选取一个商品 o_β 。如果推荐系统为商品 o_α 给出的评分大于商品 o_β 的评分，则加 1 分，如果等于则加 0.5 分。重复 n 次这个过程，记录其中出现大于情况发生的次数 n_1 ，出现等于的次数 n_2 ，则 AUC 的近似值等于：

$$AUC = \frac{n_1 + 0.5n_2}{n} \quad (2-21)$$

不难发现，AUC 指标的数值等于 0.5 意味着用户对物品的预测评分是系统随机产生的。AUC 数值越大代表算法的推荐性能越好。

2.4.2 个性化指标

(1) **多样性**。多样性^[82, 146]是用来衡量不同用户推荐列表之间的差异性。任意两个用户的推荐列表差异度越大，系统的多样性越好。给定两名用户 u_i 和 u_j ，可以用海明距离（Hamming Distance）来计算两者推荐列表之间的多样性 $HD_{ij}(L)$ ，计算公式如下：

$$HD_{ij}(L) = 1 - \frac{Q_{ij}(L)}{L} \quad (2-22)$$

其中， $Q_{ij}(L)$ 表示用户 u_i 和 u_j 在长度为 L 的推荐列表中重叠的商品个数。所有用户对之间的平均多样性即为系统整体的多样性：

$$HD(L) = \frac{1}{|U|(|U|+1)} \sum_{i \neq j} HD_{ij}(L) \quad (2-23)$$

越高的多样性代表推荐系统具有越高的个性化推荐能力。

(2) **新颖性**。新颖性是除多样性以外的又一个衡量推荐系统个性化程度的评价指标。新颖性指用户推荐列表中出现冷门物品的程度。为了定量计算推荐系统的

新颖性，一种最简单的新颖性度量方式是使用被推荐物品的平均度作为衡量标准。推荐系统对目标用户 u_i 的新颖性定义为：

$$Nov_i(L) = \frac{1}{L} \sum_{a \in L_i} k_a \quad (2-24)$$

其中， L_i 表示用户 u_i 的推荐列表。所有用户的平均新颖性即为系统整体的新颖性指标：

$$Nov(L) = \frac{1}{|U|} \sum_{i=1}^{|U|} Nov_i(L) \quad (2-25)$$

(3) 覆盖率。覆盖率指所有用户的推荐列表中独立商品占商品总数的比例，计算公式如下：

$$Cov(L) = \frac{1}{|O|} \sum_{a=1}^{|O|} \delta_a \quad (2-26)$$

其中， O 表示全部商品的集合，当物品 o_a 至少推荐给一个用户时 $\delta_a = 1$ （即至少出现在一个用户的推荐列表中时），否则等于 0。显而易见的，如果推荐系统更多的推荐流行物品则会导致覆盖率较低，低覆盖率使得用户能够通过推荐列表选择的商品数很少，这不仅会降低用户的满意度，更会使得商家不满意，因为众多的商品无法通过推荐系统推送到潜在购买者的面前，加剧了系统的马太效应。

2.5 实验数据集介绍

随着推荐系统成为研究领域的热门学科，已经有许多企业为研究人员提供了众多的公开数据集以供实验研究。在研究推荐算法性能的过程中，需要将这些数据集划分为训练集和测试集。时间感知的推荐算法是以时间信息作为上下文环境的动态推荐算法，与静态推荐算法在划分训练集和测试集时有着明显的不同。

2.5.1 公开数据集

目前可以用来进行实验研究的公开数据集已经涵盖了电影、图书、社交、餐饮、旅游等多个领域，其中一些是带有用户行为时间信息的，另一些则没有。本文主要使用了以下几个带有时间信息的公开数据集进行时间感知的推荐算法研究。

(1) Netflix 电影评分数据集。Netflix 数据集是由电影租赁公司 Netflix 在 2006 年举办的 Netflix Prize 推荐系统大赛中提供的电影评分数据集。数据集包含 40 万用户和 17000 部电影，记录了从 1999 年到 2005 年之间用户对物品的一亿条评分记录。评分区间从 1 到 5 分，其中 1 分表示非常不喜欢，5 分表示非常喜欢，分值越大表示用户越喜欢该部电影。

(2) MovieLens 电影评分数据集。MovieLens 是一个著名的电影评分系统，由美国明尼苏达大学的 GroupLens 研究小组从该网站中收集并整理了三份常用的实验数据集，该数据集进行过前期过滤以保证每名用户至少对 20 部电影进行了评分，评分范围从 1 分到 5 分。

- MovieLens 100k：是 MovieLens 数据集中最小的一份，包含从 1997 年 9 月 19 日开始至 1998 年 4 月 22 日期间 943 名用户对 1682 部电影的 10 万条评分记录。
- MovieLens 1M：该数据集包含了从 2000 年 4 月 26 日至 2003 年 3 月 1 日期间 6040 名用户对 3900 部电影的一百万条评分记录，是 MovieLens 数据集中中等大小的一份数据集。
- MovieLens 10M：是 MovieLens 数据集中最大的一份，包含从 1995 年 1 月 9 日至 2009 年 1 月 5 日期间 71567 名用户对 10681 部电影的一千万条评分记录。此外，该份数据集提供了近十万条标签数据。

(3) MovieTweetings 电影评分数据集。MovieTweetings^[147] 是一个类似 MovieLens 的电影评分数据集，该数据集中的数据是通过收集用户在 Twitter 中对电影的评分信息整理形成的，用户评分范围从 1 到 10 分。该数据集从 2013 年 2 月 28 日起开始收集，并以每天 500 条新记录的速度持续更新扩充中。该数据集包含“latest”和“snapshots”两个文件夹，其中“latest”包含最新的持续增长中的数据，而“snapshots”包含了固定大小的数据集切片。例如，10K snapshot 包含了头 1 万条从 tweets 中采集到的评分记录，20K snapshot 代表头 2 万条记录，以此类推。

(4) Delicious 书签数据集。Delicious 数据集包含来自 Delicious 书签系统中用户的社交网络、书签和标签信息。由第二届推荐系统信息异构性和融合技术国际研讨会 (HetRec 2011) 发布，具体包含从 2009 年 11 月 20 日至 2010 年 11 月 10 日期间 1867 名用户的 105000 条书签记录。

(5) FriendFeed 订阅数据集。FriendFeed 是一个实时摘要聚合工具，可以整合来自社交媒体、社交网站、社交书签网站、博客和微博等 RSS 订阅源的更新内容。FriendFeed 用户可以在该平台上聚合来自博客、Twitter、Flickr 等不同社交网络的消息内容。数据集包含从 2009 年 9 月 6 日至 2009 年 9 月 19 日期间用户对物品的喜爱信息以及用户的社交网络关系。

2.5.2 数据集划分方式

为了衡量不同推荐算法的推荐性能，研究人员需要将公开数据集划分为训练集和测试集，训练集作为历史的已知数据用来训练模型，测试集作为未知的用户

偏好对推荐算法进行性能评测。静态推荐算法和动态推荐算法在划分训练集和测试集时存在着明显的差异。

(1) 静态推荐算法的数据集划分方式。按划分的对象不同可以将数据集的划分方式分为按个体划分的和按整体划分。按个体划分是指依次将每个用户的数据按一定比例的（例如 80%）随机划入到训练集中，剩余的部分作为测试集。这种方式能够保证每位用户在训练集和测试集中都有记录，以便针对所有用户计算推荐性能。个体划分的一个特例称为留一法，仅将某个用户的一条记录作为测试集，其他全部数据作为训练集考虑。按整体划分是指将全部数据按一定比例（例如 80%）随机分配到训练集中，剩余的部分作为测试集。该方法经常与交叉验证一起使用，即每个记录作为训练集的同时也作为测试集。如图2-5所示，这是一个六折交叉验证的示意图，数据被划分为 6 等份，每次取其中一份作为测试集，其余五份作为训练集，实验结果以 6 次结果的均值作为最终结果。

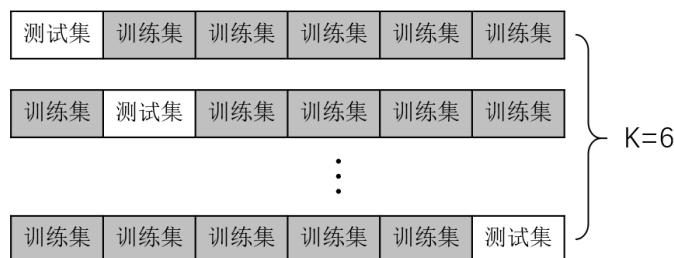


图 2-5 K 折交叉验证示意图

(2) 动态推荐算法的数据集划分方式。不同于静态推荐算法，动态推荐算法的使用场景是使用用户的历史行为来预测将来的偏好，对数据的划分有着很强的时间约束。动态推荐算法的数据集划分方式按照划分对象的不同也可以分为按个体划分和按整体划分。按个体划分是指将每个用户的行为记录按时间先后顺序排列，取距离当前时间最近的一定比例（例如 10%）数据作为给测试集，剩余的数据作为训练集。这种划分方式能够保证每个用户在测试集中都有数据可供评估。但是这种方法存在一个问题，对目标用户预测将来的兴趣选择时，推荐算法会使用到其他用户的未来行为，与真实场景存在一定区别。按整体划分是指将所有数据按照时间先后顺序排列，取距离当前时间最近的一定比例（例如 10%）或者一定时间区间内（最近一个月）的数据作为测试集，剩余的部分作为训练集。该方法的优点是能够模拟真实推荐系统的使用场景，即严格使用历史数据预测未来兴趣。但是该方法无法保证每个用户都在测试集中有记录可供评估，通常在测试集中只包含少量的系统用户。此外，可以通过滑动时间窗口的方式来生成 N 份训练集和测试集，避免只通过一份训练集导致的训练误差。如图2-6所示，每次取一个时间窗

口大小的数据作为测试集，发生在该时间窗口前特定长度的数据作为训练集。每次往后移动一个时间窗口，可以按照需求生成 N 份训练集和测试集，最后取 N 次实验的均值作为结果。



图 2-6 动态推荐算法的数据集划分方式示意图

2.6 本章小结

本章首先介绍了推荐系统在多个商业领域的应用情况以及主要面临的挑战和问题，简述了学术界研究最多的几种常用推荐算法。接着讨论了常用的离线评价指标和本文使用的带时间信息的公开数据集。最后对比了静态推荐算法和动态推荐算法划分训练集和测试集的不同之处。

第三章 基于显性评分的时间感知推荐算法研究

3.1 问题的提出

推荐系统通常利用用户与物品间的各种交互信息来产生相关的推荐结果。对于推荐算法，最基本和最方便的输入数据是用户对物品表达出的显性反馈（*explicit feedback*）。用户评分数据是典型的用户显性反馈，评分越高代表用户对物品的喜爱程度越深。例如，图3-1显示了淘宝和豆瓣电影的评分系统，用户可以用1到5颗星来表达对物品的喜爱程度。有些系统为了更加细粒度的收集用户偏好，用户的评分可以从1分到10分。随着新用户逐渐变为老用户，用户会慢慢变得沉默起来，不再明确地反馈给系统对物品具体的偏好程度。推荐系统开始根据诸如用户读过的书籍、访问过的页面、听过的歌曲、购买过的商品这类隐性反馈（*implicit feedback*）来产生推荐结果。虽然用户的隐性反馈更容易收集，但是显性评分反馈依然是衡量用户偏好最好的数据来源。



图 3-1 淘宝和豆瓣的评分系统界面

用户在阅读一本图书后可能会不喜欢该书的内容并给出较低的分数，但是用户其实对该图书中的一些信息（例如图书的题目、摘要、作者、类型等）是有较高兴趣的，否则他不会去尝试阅读该书。结果显示用户经常对他们熟悉或者感兴趣的事物表现得更加挑剔^[148]。负面评分（评分小于中间值的评分，例如在5分评价系统中，1分和2分通常被认为是负面评分）其实包含有丰富的用户兴趣信息，不应该被全盘的忽视和否定。许多推荐算法^[77, 80, 135, 149–151]只关注于用户是否购买了特定的物品，评分数据被粗粒度的处理成了二元数据。例如在文献[152]中，用户对物品的评分超过2分时，系统才认为用户对该物品是有兴趣的。在这些系统中，负面评分被认为对推荐算法起着消极的作用并被系统所忽视。这些忽视负面评价的推荐方法虽然在一般情况下能够得到较好的推荐性能，但是通过粗粒度的方式将数据集中的负面评价信息进行过滤会导致丢弃部分有用的信息，这

些算法在稀疏性很高的数据集中并不能带来很好的推荐结果。

近期，一些研究结果表明负面评分在推荐算法中也起着积极的作用，能够更好的捕捉用户行为特性并提高推荐算法的性能。Guo 等人^[150] 通过考虑用户的正面和负面评价提出了一种改进的基于用户的热传导算法。该算法分别对用户感兴趣和不感兴趣的物品进行独立地预测，然后将不会喜欢的物品从兴趣列表中剔除。结果表明该方法显著提高了推荐准确性。Zeng 等人^[148] 提出了一种加权的资源分配算法，该算法为用户的正面和负面评价赋予不同的权重，并通过一个参数来调节正面评价在算法中所占的比重。实验结果表明用户的负面评价对提高推荐准确率起着积极的作用。He 等人^[153] 提出了一种基于用户意见的协同过滤算法，该算法将用户正面和负面的评分意见融合到用户相似度计算过程中。实验结果表明该算法可以得到比基于其他相似度的协同过滤算法更好的评分预测结果。Evgeny 等人^[154] 同样通过用户的负面反馈来学习了用户的兴趣偏向，使用推荐系统为新用户提供更好的推荐结果。但是，用户正面和负面评价对用户未来兴趣表现出的时间影响力是否不同，两者对推荐算法的性能提升是否起着不同的作用，这些问题都还未被深入研究，这是本章主要的研究内容。

3.2 用户显性行为的时间特性分析

用户的显性反馈是用户有目的地主动提供给系统的输入数据，可以很明确的表达用户的兴趣爱好。在许多系统中，用户可以对自己喜欢或不喜欢的商品给予正面和负面评价，即使是负面评价也不代表用户对物品是完全不喜欢的。用户的评分标准可能随环境和时间表现得不同。例如，用户对音乐内容的评分往往比较随意，因为大部分用户在收听歌曲时，是属于没有明确目的的状态，更多的是将歌曲作为一种背景音乐；而在购物网站中的评分往往比较苛刻，因为不合格的产品会使用户感觉十分沮丧和愤怒。本章将首先对用户显性反馈行为中表现出的时间特性进行分析。为了方便本章实证分析和算法的描述，表3-1给出了本章使用的符号和标记。

3.2.1 不同用户的评分标准差异

在同一个系统中，不同的用户对内容的衡量尺度可能是不同的。有的用户习惯给商品打较高的分数，对不喜欢的商品也会打 3 分或 4 分；有的用户则喜欢明确表达自己的观点，对不喜欢的商品只打 1 分。本章首先分析电影评分数据集中不同用户评分标准的差异性。图3-2(a)、(b) 和 (c) 分别显示了 MovieLens、Netflix 和 MovieTweetings 数据集中每位用户的评分标准差分布情况。三个数据集中用户评分标准差的均值分别是 0.36427、0.56429 和 0.79849，均显著的大于 0 (单边 t 检验

表 3-1 符号和标记列表

| 符号 | 说明 |
|-------------------------------|--------------------------------------|
| u, U | 用户和用户集合 |
| o, O | 物品和物品集合 |
| S_i, S_i^k | 用户 u_i 的会话集合, 用户 u_i 的第 k 个会话 |
| r_{ia}, t_{ia} | 用户 u_i 对物品 o_a 的评分和评分时间 |
| r_i^{max}, r_i^{min} | 用户 u_i 的最大、最小评分 |
| P_i^k, N_i^k | 用户 u_i 的第 k 个会话中正面评价和负面评价相关的物品集合 |
| $r_{ik}^{max}, r_{ik}^{min}$ | 用户 u_i 的第 k 个会话中最大、最小评分 |
| λ, μ | 分别控制正面评价、负面评价衰减速率的参数 |
| $N_v, N_v \cap S, N_v \cap U$ | 网络中节点 v 的邻居集合、会话邻居集合、用户邻居集合 |
| η | 调节能量从物品节点传递到用户或会话节点比例的参数 |
| v, p | 时间段图模型中的节点和路径 |
| $w_{v,v'}$ | 路径 p 中连边 $e_{v,v'}$ 的权重 |
| $\gamma(v_0)$ | 会话节点 v_0 初始化资源 |
| β | 调节用户和会话节点初始化资源比重的参数 |

p 值接近于 0)。这表明在线系统中每个用户的评分并不是沿着系统评分区间均匀分布的, 用户有着自己的评分标准。从图中可以发现, 绝大多数用户评分的标准差集中在 0.3 到 0.5 分之间, 并不会明确表现出对物品的喜爱。但也有一些用户的评分标准差超过了 2 分, 尤其是在 MovieTweetings 这种 10 分的细粒度评分系统中。

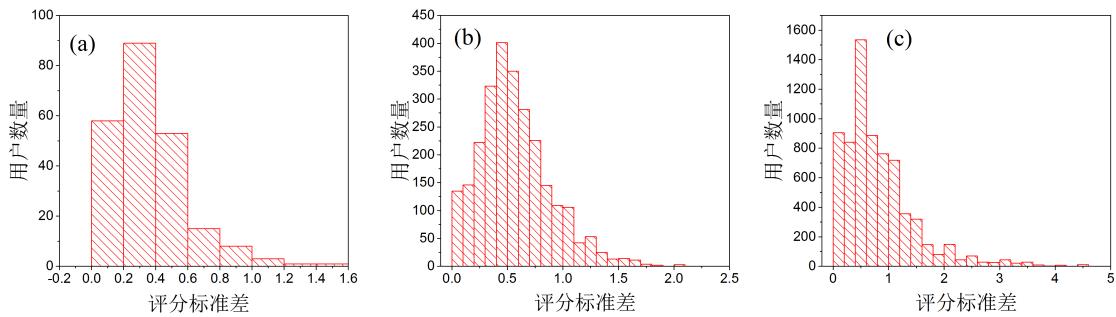


图 3-2 用户评分标准差的分布情况。(a) MovieLens; (b) Netflix; (c) MovieTweetings

为了将不同用户的评分信息在统一的尺度下进行比较, 本章使用如下公式对用户原始评分进行了归一化, 将每个用户对物品的评分统一到 $[0,1]$ 的区间内:

$$R_{ia} = \frac{r_{ia} - r_i^{min}}{r_i^{max} - r_i^{min}} \quad (3-1)$$

会话（Session）表示用户活动的一段连续时间，例如，用户 u_i 在今年 9 月份的活动可以理解成用户 u_i 的一个会话。根据会话的定义，用户 u_i 的行为可以切分成一系列会话的有序集合 $S_i = \{S_i^1, S_i^2, \dots, S_i^T\}$ 。本章以自然月为会话的时间长度，将用户的行为按自然月进行切分。当且仅当两个评分记录是由同一用户在同一个自然月中产生时，这两个评分记录才会被分配到同一个会话中。用户的兴趣会随着时间发生漂移，不同时间段内的评分尺度也可能发生变化。图3-3给出了 MovieLens、Netflix 和 MovieTweetings 数据集上每个用户在第一个月和最后一个月的平均评分对应关系。从图中可以发现，虽然每个用户在这两个月的平均评分关系都集中在对角线附近，但是依然有很多用户在这两个月的评分标准表现出很大的差异性。为了考虑用户在不同时间段表现出的评分标准差异，本章将用户在特定会话 S_i^k 的评分根据该会话中的最大最小评分值进行了归一化处理以便在同一尺度中进行比较：

$$R_{i\alpha}^k = \frac{r_{i\alpha} - r_{ik}^{\min}}{r_{ik}^{\max} - r_{ik}^{\min}} \quad (3-2)$$

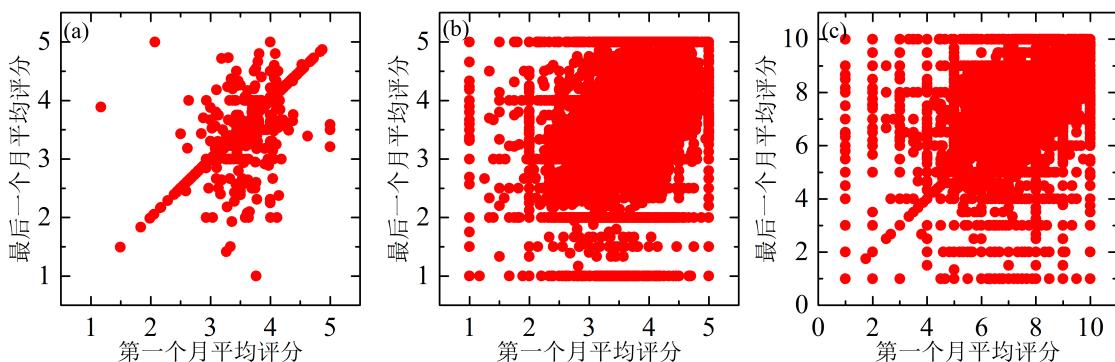


图 3-3 用户在第一个月和最后一个月平均评分的对应关系。(a) MovieLens; (b) Netflix;
(c) MovieTweetings

3.2.2 正负面评价的时间特性差异

用户对物品的评分信息可以被分为对物品的正面评价和负面评价。例如在 1-5 分的评分系统中，如果用户评分小于 3 分，则认为用户对物品给予了一个负面评价，否则认为这是一个正面评价。本章接着研究用户的正负面评价对用户将来有兴趣是否具有不同的时间影响力。

针对用户 u_i ，将该用户的第 k 个会话 S_i^k 中正面评价的相关物品组成的集合记为 P_i^k 。类似的，用户 u_i 的第 k 个会话中负面评价的相关物品组成的集合记为 N_i^k 。图3-4给出了用户 u_i 的评分数据按会话进行划分后，各会话中的正面评价和负面评

价关联的物品集合组成的序列。

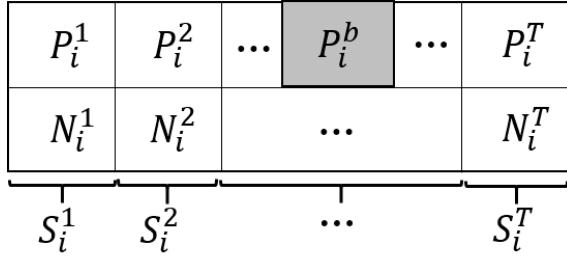


图 3-4 会话中正面评价物和负面评价关联的物品组成的集合序列

集合 X 与集合 Y 之间的相关性定义为集合 X 和集合 Y 之间每一对物品的平均余弦相似度，记为 $\text{Cor}(X, Y)$ 。对于任意一个会话 S_i^k ，可以将其中的 P_i^k 作为用户当前时刻的兴趣爱好，而所有发生在 S_i^1 到 S_i^{k-1} 会话中的正面和负面评价相关的物品作为该用户的历史偏好。因此， $\text{Cor}(P_i^a, P_i^b), a < b$ 可以表示为用户过去正面评价的物品与未来兴趣间的相关性；类似的， $\text{Cor}(N_i^a, P_i^b), a < b$ 表示用户过去负面评价的物品与未来兴趣间的相关性。为了分析用户历史的正负面评价对未来兴趣的不同时序影响力，本章首先计算同一个用户的每两个会话间的 $\text{Cor}(P_i^a, P_i^b)$ 相关性与会话时间间隔 $\Delta t = t_b - t_a$ 的对应关系，然后对该用户相同 Δt 对应的 $\text{Cor}(P_i^a, P_i^b)$ 计算了均值，得到该用户在不同的时间间隔下正面评价与未来兴趣的平均相关性。类似的，可以通过 Δt 与 $\text{Cor}(N_i^a, P_i^b)$ 的对应关系得到每个用户在不同的时间间隔下负面评价与未来兴趣的平均相关性。为了方便描述，本章将根据所有用户求均值得到 Δt 与 $\text{Cor}(P_i^a, P_i^b)$ 和 $\text{Cor}(N_i^a, P_i^b)$ 的对应关系曲线称为正负面评价的时间效应曲线。图3-5(a)、(b) 和 (c) 分别显示了在 MovieLens、Netflix 和 MovieTweetings 数据集上的正负面评价时间效应曲线。为了证明实验结果的真实性，本章将每个用户行为的评分时间进行了洗牌，打乱了原有的行为顺序，并按照相同的方法研究了在洗牌后的数据上 Δt 与 $\text{Cor}(P_i^a, P_i^b)$ 和 $\text{Cor}(N_i^a, P_i^b)$ 的对应关系。图3-5中， CorP 代表过去正面评价对未来兴趣的影响、 CorN 代表过去负面评价对未来兴趣的影响， $\text{CorP} - R$ 和 $\text{CorN} - R$ 是对用户行为时间进行随机洗牌 100 次得到的平均结果。

从图3-5中可以发现用户历史的正面评价和负面评价与未来的兴趣偏好都有着较强的相关性，正面评价的相关性要高于负面评价的。此外，无论用户的正面评价还是负面评价都存在随时间衰减的现象，近期的评价行为对用户将来兴趣的影响力更强。不同数据集中，正面评价和负面评价的衰减速率不尽相同。在将用户行为时间进行随机洗牌后，用户历史正负面评价与未来兴趣间的相关性明显降低，同时，正负面评价与未来兴趣相关性的时间衰减程度也没有真实数据中那么明显。这表明在真实数据集中用户正负面评价与未来兴趣的相关时间特性是客观

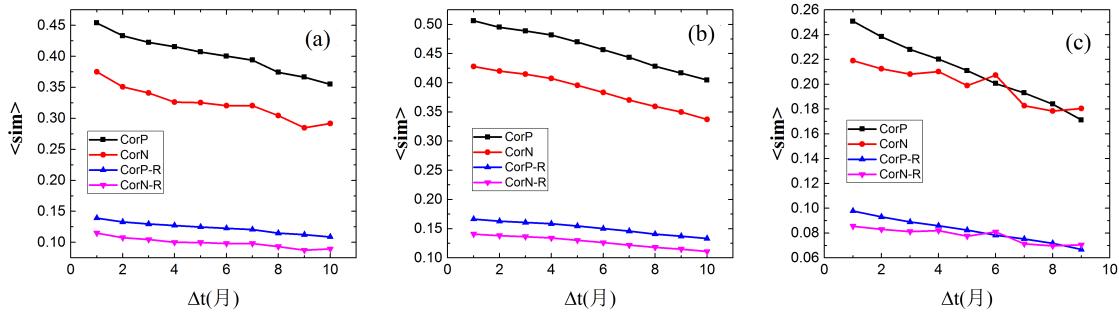


图 3-5 (a) MovieLens、(b) Netflix 和 (c) MovieTweetings 数据集上的 $CorP$ 、 $CorN$ 、 $CorP-R$ 和 $CorN-R$ 曲线

存在的，对时间的随机扰动会破坏这种时间特性。

3.3 正负面评价时序特性在推荐算法中的应用研究

按用户行为的时间信息将记录划分到不同的会话中，带有时间信息的推荐系统可以使用用户-物品-会话的三部图 $G(U, O, S, E)$ 进行建模，其中包含一个用户集合 $U = \{u_1, u_2, \dots, u_m\}$ 、一个物品集合 $O = \{o_1, o_2, \dots, o_n\}$ 以及一个会话集合 $S = \{S_1^1, S_1^2, \dots, S_1^T, S_2^1, \dots, S_m^T\}$ 。这三种节点一共组成了 4 类带权的有向连边，分别是 E_{UO} 、 E_{OU} 、 E_{SO} 和 E_{Os} ，前两个表示用户节点与物品节点间的双向连边，后两个表示物品节点与会话节点间的连边。这种用户-物品-会话三部图称为时间段图模型（Session-based Temporal Graph，简称 STG）^[91]，其中与会话关联的连边反映了用户在该会话时间段内的短期兴趣，与用户相连的连边反映了用户的长期兴趣。图3-6给出了一个 STG 的样例。为了方便的区分用户和物品，本章使用拉丁字母代表用户，希腊字母代表物品。

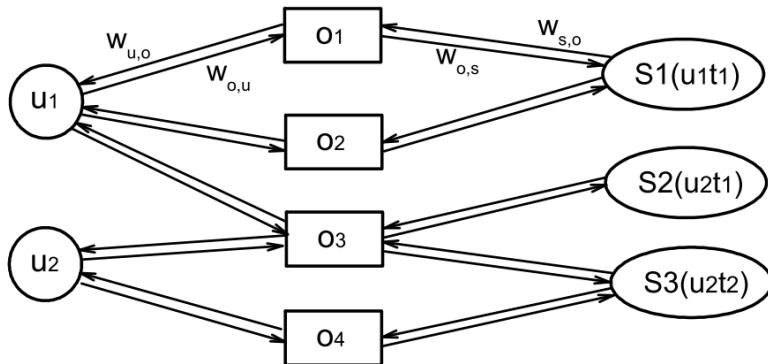


图 3-6 时间段图模型样例

3.3.1 基于正负面评价的时间感知推荐算法

从实证分析可以发现越接近目标时间 T 的评分行为对推荐任务越重要，用户正负面评价与未来兴趣相关性的时间衰减速率随不同的数据集而表现不同，因此本章为正面和负面评价行为分别赋予了不同的时间衰减因子。在 STG 图中，从用户节点到物品节点和从物品节点到用户节点的连边权重定义为：

$$w_{i,a} = w_{a,i} = \begin{cases} R_{ia} \times e^{-\lambda(T-t_{ia})}, & \text{如果 } R_{ia} \geq 0.5 \\ R_{ia} \times e^{-\mu(T-t_{ia})}, & \text{否则} \end{cases} \quad (3-3)$$

其中， λ 是正面评价的衰减因子， μ 是负面评价的衰减因子， R_{ia} 表示用户 u_i 对物品 o_a 的评分， t_{ia} 表示用户 u_i 对物品 o_a 进行评分的时间， T 表示进行预测的当前时间。类似的，在给定的会话中，距离目标时间 T 越近的记录也越重要。因此为会话中的正面和负面评价赋予与公式 (3-3) 相同的两个时间衰减因子。在 STG 图中，从物品节点到会话节点和从会话节点到物品节点的连边权重定义为：

$$w_{S_i^k,a} = w_{a,S_i^k} = \begin{cases} R_{ia}^k \times e^{-\lambda(T-t_{ia})}, & \text{如果 } R_{ia}^k \geq 0.5 \\ R_{ia}^k \times e^{-\mu(T-t_{ia})}, & \text{否则} \end{cases} \quad (3-4)$$

在对 STG 图中的连边都赋予权重后，可以将 STG 图中的连边权重归一化为：

$$\psi(v, v') = \begin{cases} \frac{w_{v,v'}}{\sum_{x \in N_v} w_{v,x}}, & \text{如果 } v \in U \cup S \text{ 且 } v' \in O \\ \frac{\eta w_{v,v'}}{\eta \sum_{x \in N_v \cap U} w_{v,x} + \sum_{x \in N_v \cap S} w_{v,x}}, & \text{如果 } v \in O \text{ 且 } v' \in U \\ \frac{w_{v,v'}}{\eta \sum_{x \in N_v \cap U} w_{v,x} + \sum_{x \in N_v \cap S} w_{v,x}}, & \text{如果 } v \in O \text{ 且 } v' \in S \end{cases} \quad (3-5)$$

其中， $w_{v,v'}$ 表示图中节点 v 到节点 v' 的连边权重，参数 η 值越大代表用户的长期兴趣在偏好传递过程中扮演着越重要的角色。

给定目标用户 u_i ，首先为用户节点 u_i 和该用户的最后一个会话节点注入初始资源，然后偏好通过 STG 图中的多条路径传递到候选物品上。每条路径 p 传播的偏好资源等于目标用户节点 u_i （或最后一个会话节点 S_i^T ）的初始资源 $\gamma(v_0)$ 与路径中每条连边的权重的乘积，记为路径 p 的权重 $\varphi(p)$ ，公式定义为：

$$\varphi(p) = \prod_{e_{v,v'} \in p} \psi(v, v') \gamma(v_0) \quad (3-6)$$

其中， $e_{v,v'}$ 表示路径 p 上节点 v 与节点 v' 之间的连边， $\gamma(v_0)$ 的数值依赖于节点 v_0

的类型:

$$\gamma(v_0) = \begin{cases} \beta, & \text{如果 } v_0 = u_i \\ 1 - \beta, & \text{如果 } v_0 = S_i^T \end{cases} \quad (3-7)$$

其中, $\beta = 0$ 表示没有偏好注入到用户节点, 偏好的扩散过程仅从用户的短期兴趣开始考虑; 而 $\beta = 1$ 表示没有偏好注入到会话节点, 偏好的扩散过程仅从用户的长期兴趣开始考虑。

对于每一个目标用户, 仅考虑从起始节点到未购买的物品节点距离等于 3 的最短路径, 使用广度搜索算法可以得到四类满足条件的路径, 分别是:

$$\left\{ \begin{array}{l} \text{路径 1: 用户-物品-用户-物品} \\ \text{路径 2: 用户-物品-会话-物品} \\ \text{路径 3: 会话-物品-会话-物品} \\ \text{路径 4: 会话-物品-用户-物品} \end{array} \right.$$

定义 P_{ia} 表示对于目标用户 u_i 从初始节点到未购买的物品节点 o_a 的最短路径集合, 用户 u_i 对物品 o_a 的偏好定义为:

$$\varphi_{ia} = \sum_{p \in P_{ia}} \varphi(p) \quad (3-8)$$

其中, $\varphi(p)$ 表示路径 p 的权重。将目标用户未购买过的物品得到的最终资源降序排序后, 排名靠前的物品被推荐给目标用户。本章称该算法为基于正负面评价的时间感知推荐算法 (Time-aware Algorithm based on Positive and Negative Ratings), 记为 TaPNR。

3.3.2 基于曲线拟合的时间感知推荐算法

TaPNR 算法共包含 4 个参数, 其中 λ 和 μ 分别用来控制正面评价和负面评价的时间衰减速率, η 和 β 用来控制用户长期偏好和短期偏好在算法中的作用。这 4 个参数的选定需要经过反复的遍历过程, 计算复杂度很高, 这对将该算法应用到实际系统中带来了极大的困难。本章主要研究的内容是分析用户显性反馈中正面和负面评价对推荐算法的不同时序作用, 因此暂时将用户长期和短期兴趣设置为相同的权重, 即将参数 η 和 β 分别设置为 1 和 0.5, 以忽略长短期兴趣对推荐算法的影响。在研究用户的历史正面评价和负面评价与未来兴趣相关性的时候, 我们绘制了数据集的正负面评价时间效应曲线。由于曲线变化相对平稳, 本章利用一

元线性方程对其进行近似拟合，拟合公式定义为：

$$\hat{s}_P = a_P t + b_P \quad (3-9)$$

$$\hat{s}_N = a_N t + b_N \quad (3-10)$$

其中， a_P 和 b_P 分别是对正面评价时间效应曲线近似拟合的直线方程的斜率和截距， a_N 和 b_N 分别是对负面评价时间效应曲线近似拟合的直线方程的斜率和截距。为了确定这 4 个系数，本章采用最小二乘法以误差平方和最小为目标进行参数估计：

$$Q_P = \sum_{i=1}^T (s_P - \hat{s}_P)^2 = \sum_{i=1}^T (s_P - a_P t_i - b_P)^2 \quad (3-11)$$

$$Q_N = \sum_{i=1}^T (s_N - \hat{s}_N)^2 = \sum_{i=1}^T (s_N - a_N t_i - b_N)^2 \quad (3-12)$$

为使 Q_P 最小，根据中心极限定理，对参数 a_P 和 b_P 分别求偏导并令其等于 0，可以得到：

$$\begin{cases} a_P = \bar{s}_P - b_P \bar{t} \\ b_P = \frac{\sum s_P^i t_i - \bar{t} \sum s_P^i}{\sum s_P^{i^2} - \bar{t} \sum t_i} \end{cases} \quad (3-13)$$

其中， $\bar{s}_P = \sum s_P^i / T$, $\bar{t} = \sum t_i / T$ 。同样的方式可以求得参数 a_N 和 b_N :

$$\begin{cases} a_N = \bar{s}_N - b_N \bar{t} \\ b_N = \frac{\sum s_N^i t_i - \bar{t} \sum s_N^i}{\sum s_N^{i^2} - \bar{t} \sum t_i} \end{cases} \quad (3-14)$$

其中， $\bar{s}_N = \sum s_N^i / T$ 。

然后利用对历史正面评价和负面评价与未来兴趣相关性的近似拟合结果对时间段图模型中的连边权重进行了重新定义：

$$w_{i,\alpha} = w_{\alpha,i} = \begin{cases} R_{i\alpha} \times [a_P(T - t_{i\alpha}) + b_P], & \text{如果 } R_{i\alpha} \geq 0.5 \\ R_{i\alpha} \times [a_N(T - t_{i\alpha}) + b_N], & \text{否则} \end{cases} \quad (3-15)$$

$$w_{S_i^k, \alpha} = w_{\alpha, S_i^k} = \begin{cases} R_{i\alpha}^k \times [a_P(T - t_{i\alpha}) + b_P], & \text{如果 } R_{i\alpha}^k \geq 0.5 \\ R_{i\alpha}^k \times [a_N(T - t_{i\alpha}) + b_N], & \text{否则} \end{cases} \quad (3-16)$$

得到连边权重后，推荐过程同基于正负面时序特性的推荐算法一样，首先为

目标用户和该用户的最近会话赋予初始资源，资源通过长度等于 3 的路径传递到候选物品节点，最终根据物品节点获得的资源数量产生推荐结果。由于利用线性回归方法拟合得到了正负面评价的不同时间衰减因子，不再需要对算法参数进行遍历寻优，本章称该算法是基于曲线拟合的时间感知推荐算法（Time-aware Recommendation based on Curve-fitting），记为 TaCF。

3.4 算法性能评测

3.4.1 实验数据集和对比算法

本章使用了两组数据，第一组数据的数据量较为丰富，主要用来进行实证分析和验证基于曲线拟合的时间感知推荐算法 TaCF 的推荐性能。该组数据包含三个电影评分数数据集，分别是 MovieLens、Netflix 和 MovieTweetings。其中 MovieLens 数据集是 MovieLens 1M 数据集从 2000 年 4 月至 2001 年 3 月期间的 942004 条评分记录。MovieTweetings 采用数据量为 200K 的 snapshot，由于原始数据中 2013 年 2 月和 2014 年 1 月的评分数信息很少，因此抽取了 2013 年 3 月至 2014 年 2 月期间的数据作为此次 MovieTweetings 的数据集，其中包含 24530 个用户和 14458 个物品，共 192360 条评分记录。Netflix 数据集^①是从原始数据集 2005 年 1 月至 2005 年 12 月期间随机抽取的 10000 名用户的评分记录，这些用户至少在 2005 年中有 6 个自然月的评分数据（本文对 Netflix 数据集进行了 10 次随机采样，算法在不同的随机数据集上得到的算法性能差异很小，因此本文对 Netflix 数据集仅列举了其中一个抽样数据集进行性能说明）。表3-2给出了第一组数据集的基本统计信息。

表 3-2 第一组数据集基本统计信息

| 数据集 | 用户数 | 商品数 | 记录数 | 稀疏性 | 评分范围 |
|----------------|-------|-------|---------|-----------------------|------|
| MovieLens | 6035 | 3687 | 942004 | 4.23×10^{-2} | 1-5 |
| Netflix | 10000 | 16885 | 1717497 | 1.02×10^{-2} | 1-5 |
| MovieTweetings | 34530 | 14458 | 192360 | 3.85×10^{-4} | 1-10 |

第二组数据的数据量较小，由于基于正负面评价的时间感知推荐算法 TaPNR 的参数确定需要采用遍历寻优的方式，使用第一组数据的计算效率很低。因此，本章采用了第二组数据来研究 TaPNR 推荐模型中各参数的作用。该组数据同样包含 MovieLens、Netflix 和 MovieTweetings 三个电影评分数数据集。其中 MovieLens 采用了 MovieLens 中最小的 100k 数据集；Netflix 数据集^②是从原始数据集 2005 年 1 月

① 可从 https://github.com/wjleeUESTC/thesis_data/tree/master/data3-1 下载

② 可从 https://github.com/wjleeUESTC/thesis_data/tree/master/data3-2 下载

至 2005 年 5 月的数据中随机抽取了 3000 名用户和 3000 部电影，共 97367 条记录。MovieTweetings 数据集使用了数据量为 100K 的 snapshot，从中剔除了评分数量少于 10 条的用户，最终包含从 2013 年 3 月至 2013 年 8 月期间 2583 名用户和 9111 部电影组成的 67040 条评分记录。表3-3给出了第二组数据集的基本统计信息。

本章将数据按照评分行为的时间进行排序，最后一个自然月中的数据作为测试集，剩余的历史数据作为训练集，同时训练集中未出现过的用户会从测试集中剔除。本章从准确性和个性化两个角度对推荐算法的性能进行评估，其中准确性使用了 F1 指标进行衡量，多样性 (HD)、新颖性和覆盖率三个指标被用来度量推荐算法的个性化水平。除了新颖性指标是越小越好，其他的评价指标均是越大越好。实验过程中算法为每位用户推荐 10 个物品。

表 3-3 第二组数据集基本统计信息

| 数据集 | 用户数 | 商品数 | 记录数 | 稀疏性 | 评分范围 |
|----------------|------|------|--------|-----------------------|------|
| MovieLens | 943 | 1682 | 100000 | 6.30×10^{-2} | 1-5 |
| Netflix | 3000 | 3000 | 97367 | 1.08×10^{-2} | 1-5 |
| MovieTweetings | 2593 | 9111 | 67040 | 2.85×10^{-3} | 1-10 |

为了分析 TaPNR 和 TaCF 算法的推荐性能，本章将两者与 UCF^[69]、UOS^[153]、NBI^[77]、SNBI^[148]、IPF^[91] 五个算法进行了比较。UOS 是一种改进的协同过滤算法，通过用户-物品网络中的正负面评价来衡量用户间的相似度。NBI 算法是一个基于网络推论的资源分配算法，比传统协同过滤算法更加高效并能提供更加准确的推荐结果。SNBI 在 NBI 的基础上将网络中的连边根据评分的正负给予不同的权重来优化计算过程。IPF 是一个基于时间段图模型的算法，将用户短期的兴趣构成一个会话，以此综合了用户的长期兴趣和短期兴趣来提高推荐准确性。此外，本章还比较了时间相关的矩阵分解算法^[89]，该方法需要训练时间相关的隐因子向量，虽然该算法在 Netflix Prize 问题下是一个良好的解决方案，但是由于本章面向的问题是严格利用用户的历史数据来预测未来的偏好，基于矩阵分解的算法无法构建满足要求的时间向量（需要的时间长度 Δt 未在训练集中出现过），导致三个数据集上最终的 F1 指标均未超过 0.001，因此本章实验结果部分忽略了该方法。对比算法的具体描述见表3-4，其中 PN 表示算法是否将用户评分意见分类为正面评价和负面评价，T 表示算法是否是时间相关的。

3.4.2 基于正负面评价的时间感知推荐算法性能分析

本章首先研究基于正负面评价的时间感知推荐算法 TaPNR 中用于控制正面和负面评价时间衰减速率的参数 λ 和 μ 对推荐性能的影响。这两个参数值越大，历

表 3-4 对比算法说明

| 方法 | 说明 | PN | T |
|------|---------------------|----|---|
| UCF | 基于用户的协同过滤算法 | - | - |
| UOS | 结合了用户正负面意见的 UCF 算法 | ✓ | - |
| NBI | 基于 0-1 数据的网络推论算法 | - | - |
| SNBI | 赋予正负面评分不同权重的 NBI 算法 | ✓ | - |
| IPF | 基于时间段图模型的个性化随机游走算法 | - | ✓ |

史评价对将来兴趣的影响力反而越小。不失一般性的，在调节参数 λ 和 μ 时，把 η 设置成 1， β 设置成 0.5，以此忽略这两个参数对推荐算法的影响。

图3-7给出了在 MovieLens、Netflix 和 MovieTweetings 三个数据集上，TaPNR 算法推荐结果的 F1 指标随参数 λ 和 μ 的变化情况。在图3-7中，不同的颜色代表不同的推荐准确性，深红色表示准确性最大的值，深蓝色代表准确性最小的值。首先，从图中可以观察到，相较于参数 λ ，F1 指标沿 μ 轴的变化范围要更大，表明 TaPNR 算法推荐性能对用户负面评价更加敏感。例如，当用户购买了一件某品牌的商品发现质量严重不合格时，很大可能不会对该品牌产生忠诚度。其次，在所有三个数据集中都可以发现平行于 λ 抽的“山脊”，TaPNR 算法在“山脊”附近可以得到全局最优的 F1 指标。因此，为了快速发现全局最优 F1 指标，可以先将 λ 设置在 $10^{-3} \sim 10^{-2}$ 区间中的一个较小值，通过先调节参数 μ 得到局部最优值，然后根据局部最优的 μ 值再调节参数 λ 得到全局最优 F1 指标。

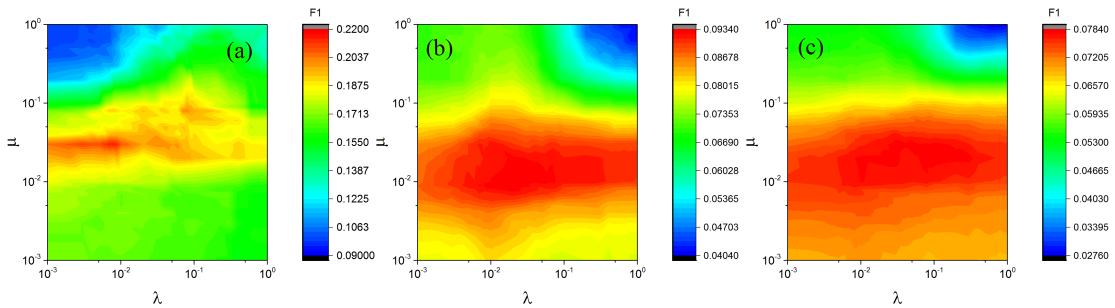


图 3-7 TaPNR 算法 F1 指标与参数 λ 和 μ 的关系。(a) MovieLens; (b) Netflix; (c) MovieTweetings

当参数 λ 和 μ 都等于 0 时，TaPNR 算法中的时间作用将被忽略，表3-5列出了 TaPNR 算法在考虑及不考虑时间因素时得到的推荐结果，包括 F1 指标、多样性、新颖性和覆盖率。通过表中结果可以发现，除了多样性指标以外，考虑了正负面时间特性的推荐准确性、新颖性和覆盖率都要明显好于不考虑时的结果，而损失

的多样性指标相对比例也很小，这说明时间因素在该算法中起着很大的积极作用。此外，在 MovieLens 数据集中，正面评价的时间衰减因子 λ 要明显小于负面评价的衰减因子 μ ，而在 MovieTweetings 数据集中参数 λ 却要大于参数 μ ，在 Netflix 数据集中参数 λ 和参数 μ 几乎是一致的。为了分析这个现象，本章分析了该组数据集中的 CorP 和 CorN 曲线，分析结果见图3-8。从图中可以发现，在 MovieLens 数据集中 CorP 的衰减速率要小于 CorN 的，在 Netflix 数据集中两者基本一致，而在 MovieTweetings 数据集中 CorN 的衰减速率要比 CorP 大得多。这些现象与在调节参数过程中发现的基本一致，说明这两个参数一定程度上可以客观反映系统中用户行为的时间特性。

表 3-5 TaPNR 算法考虑和不考虑时间因素时的推荐结果

| 数据集 | λ | μ | F1 | 多样性 | 新颖性 | 覆盖率 |
|----------------|-----------|--------|---------------|---------------|---------------|---------------|
| MovieLens | 0.0 | 0.0 | 0.1697 | 0.7946 | 306.37 | 0.1058 |
| | 0.007 | 0.03 | 0.2148 | 0.7647 | 275.18 | 0.1285 |
| Netflix | 0.0 | 0.0 | 0.0770 | 0.6848 | 600.66 | 0.2976 |
| | 0.0009 | 0.0009 | 0.0930 | 0.6757 | 555.41 | 0.3067 |
| MovieTweetings | 0.0 | 0.0 | 0.0689 | 0.6869 | 381.43 | 0.2084 |
| | 0.09 | 0.02 | 0.0783 | 0.6582 | 344.52 | 0.2100 |

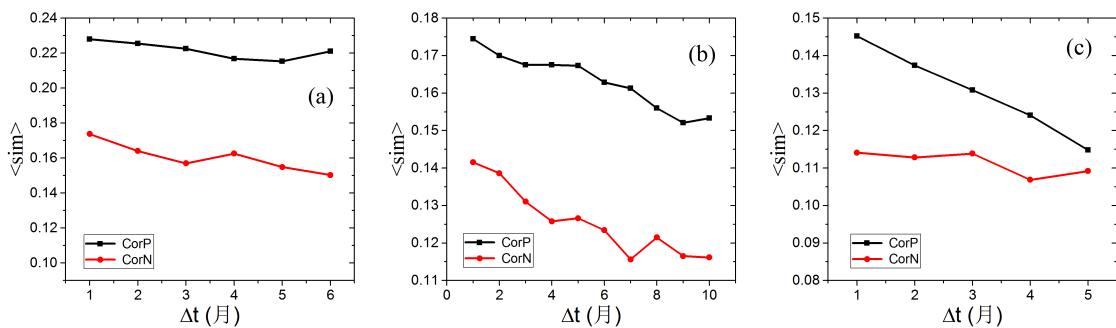


图 3-8 历史评价与未来偏好间的相似度随时间间隔的变化关系。(a) MovieLens; (b) Netflix; (c) MovieTweetings

在 TaPNR 模型中，参数 β 用来控制注入到用户节点和会话节点的初始资源，在一定程度上反映了用户短期和长期兴趣所占的比重。如果 $\beta = 0$ ，则没有能量注入到用户节点，而当 $\beta = 1$ 时，没有能量注入到会话节点。 β 值越，长期兴趣占的比重越多。图3-9给出了三个数据集中推荐准确性与参数 β 之间的关系。首先，从结果中可以发现忽略长期兴趣（即把 β 设置为 0）是无法得到最优推荐结果的。其次，在稀疏性较大的数据集上（稀疏性排名：MovieTweetings>Netflix>MovieLens），

最优 F1 指标对应的参数 β 值往往偏大。从用户长期兴趣出发可以更好的发现用户未来的兴趣，从短期兴趣出发可能会因为数据的稀疏性带来偶然性使得会话中的兴趣与用户真实偏好有一定偏差。一般而言，最优推荐结果需要同时考虑用户的长期和短期兴趣。在讨论参数 η 与推荐结果关系时，MovieLens、Netflix 和 MovieTweetings 三个数据集上对应的参数 β 分别设置为 0.5、0.9 和 1.0。

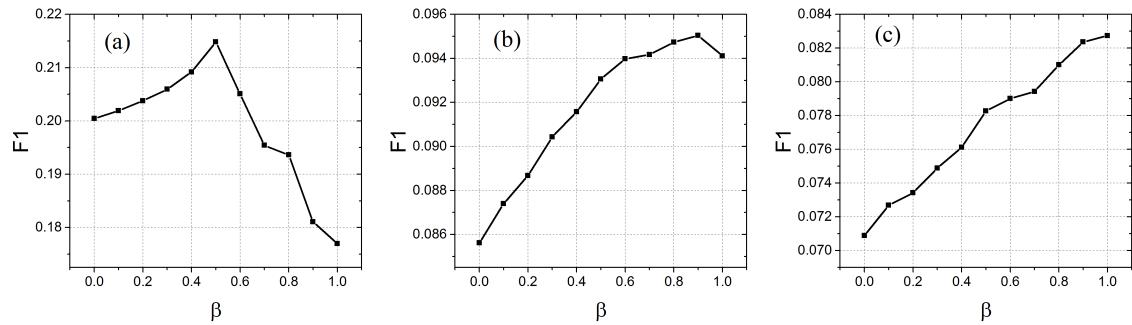


图 3-9 TaPNR 推荐准确性与参数 β 的关系。(a) MovieLens; (b) Netflix; (c) MovieTweetings

与参数 β 类似，参数 η 也是用来平衡长期和短期兴趣，不同之处在于参数 η 的控制作用发生在资源扩散过程中。当 $\eta = 0$ 时，物品节点只与用户节点相连，物品与物品之间的相似度仅依赖于用户的长期兴趣决定；当 η 趋于无穷大时，物品节点只与会话节点相连，物品间的相似度仅依赖于用户的短期兴趣。图3-10给出了三个数据集上 F1 指标与参数 η 的关系。在 MovieLens 和 Netflix 数据集上，最优 F1 指标对应的 η 值分别在 1 和 10 附近，而 MovieTweetings 数据集中最优的 η 值在 100 附近。结果表明用户的长期和短期兴趣对于衡量物品间的相似度都是十分重要的因素。更进一步的，在 MovieLens、Netflix 和 MovieTweetings 数据集上，随着数据集越来越稀疏，参数 η 的值越来越大，用户长期兴趣的作用要逐渐大于短期兴趣，这与参数 β 的情况相似。

通过算法性能分析发现，想要得到较高的推荐性能，TaPNR 算法需要同时考虑用户长期和短期兴趣，不能忽略其中任何一个。在稀疏数据集上，用户长期兴趣的重要性要大于短期兴趣。当把参数 η 和 β 设置为 1 和 0.5 时，系统中的长期和短期兴趣提供了相同的重要性，此时推荐结果的准确性依然是较高的。这对 TaCF 算法将用户长期和短期作用设置为相同权重提供了一定的数据支持。

3.4.3 不同算法的推荐性能比较

为了进一步说明用户显性反馈中正负面评价对于提高推荐系统性能方面的作用，本章将 TaPNR 推荐模型与几个基准模型进行了对比。表3-6、表3-7和表3-8分

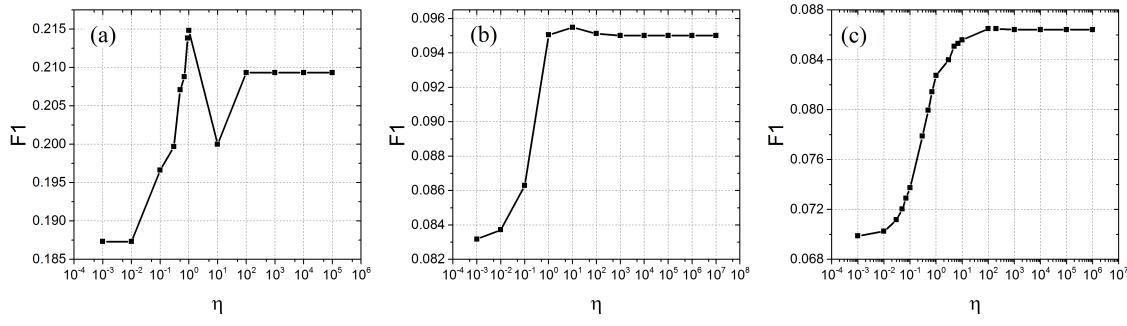


图 3-10 TaPNR 推荐准确性与参数 η 的关系。(a) MovieLens; (b) Netflix; (c) MovieTweetings

别给出了 MovieLens、Netflix 和 MovieTweetings 三个数据集上六个推荐算法的推荐性能对比结果。从三张表的结果可以发现：相较于基于网络结构的算法（NBI 和 SNBI），基于协同过滤的算法（UCF 和 UOS）可以得到更多样、更新颖的推荐结果，同时被推荐的物品的覆盖率也较高，但是推荐准确性要低很多；在考虑将用户评分划分为正面和负面评分后，改进的协同过滤算法（UOS）和基于网络结构的算法（SNBI）都比原始算法（UCF 和 NBI）提高了推荐准确性，UOS 和 SNBI 算法的 F1 指标要分别明显优于 UCF 和 NBI 算法。在多样性、新颖性和覆盖率方面，SNBI 也要比 NBI 算法有所提高。但是在稀疏数据集 MovieTweetings 上，UOS 和 SNBI 算法的结果不如预期那样，SNBI 的结果基本跟 NBI 相同，而 UOS 的 F1 指标要比 UCF 更差；除了在 Netflix 数据集上，考虑了长期和短期兴趣的 IPF 算法可以提供比 UCF、UOS、NBI 和 SNBI 更好的推荐结果；本章提出的 TaPNR 算法能够提供比其他算法更高的推荐准确性。除了在准确率方面，针对稀疏数据集 MovieTweetings，TaPNR 算法能够提供更好的推荐多样性、新颖性和覆盖率。

表 3-6 MovieLens 数据集上不同算法的推荐性能对比

| 算法 | F1 | 多样性 | 新颖性 | 覆盖率 |
|--|---------------|---------------|---------------|---------------|
| UCF | 0.0761 | 0.9399 | 224.63 | 0.2852 |
| UOS($\lambda = 0.2$) | 0.0839 | 0.8735 | 268.10 | 0.1691 |
| NBI | 0.1081 | 0.7724 | 319.62 | 0.0873 |
| SNBI ($\lambda = -2.1$) | 0.1144 | 0.8382 | 287.57 | 0.1353 |
| IPF($\beta = 0.1$, $\rho = 1.5$, $\eta = 1.0$) | 0.1649 | 0.8010 | 281.11 | 0.1667 |
| TaPNR($\beta = 0.5$, $\lambda = 0.007$, $\mu = 0.03$, $\eta = 1.0$) | 0.2147 | 0.7647 | 275.18 | 0.1285 |

表 3-7 Netflix 数据集上不同算法的推荐性能对比

| 算法 | F1 | 多样性 | 新颖性 | 覆盖率 |
|---|---------------|---------------|---------------|---------------|
| UCF | 0.0732 | 0.8876 | 453.55 | 0.2720 |
| UOS($\lambda = 0.0$) | 0.0733 | 0.7638 | 546.90 | 0.1666 |
| NBI | 0.0752 | 0.5908 | 677.00 | 0.0942 |
| SNBI($\lambda = -1.3$) | 0.0794 | 0.6057 | 657.37 | 0.1441 |
| IPF($\beta = 0.5, \rho = 1.1, \eta = 10$) | 0.0783 | 0.6607 | 636.08 | 0.2084 |
| TaPNR($\beta = 0.9, \lambda = 0.009, \mu = 0.009, \eta = 10$) | 0.0950 | 0.6169 | 589.88 | 0.2358 |

表 3-8 MovieTweetings 数据集上不同算法的推荐性能对比

| 算法 | F1 | 多样性 | 新颖性 | 覆盖率 |
|--|---------------|---------------|---------------|---------------|
| UCF | 0.0633 | 0.7966 | 355.17 | 0.1185 |
| UOS($\lambda = -0.2$) | 0.0490 | 0.7785 | 337.87 | 0.1837 |
| NBI | 0.0706 | 0.4881 | 468.30 | 0.0440 |
| SNBI($\lambda = -0.9$) | 0.0725 | 0.4999 | 463.43 | 0.0542 |
| IPF($\beta = 0.1, \rho = 1.5, \eta = 1.0$) | 0.0785 | 0.5024 | 462.35 | 0.0447 |
| TaPNR($\beta = 1.0, \lambda = 0.09, \mu = 0.02, \eta = 100$) | 0.0865 | 0.5269 | 394.51 | 0.1341 |

3.4.4 基于曲线拟合的时间感知推荐算法性能分析

TaCF 推荐算法在得到较高的计算效率时，忽略了用户长期和短期兴趣不同的作用，给予他们相同的权重。同时，为了避免对参数的遍历寻优，利用线性回归方式对正负面评价与未来兴趣的相关曲线进行了近似拟合。接下来，本章将研究这种近似求解方式是否可以得到较好的推荐性能。本节实验部分使用了第一组数据量较多的数据用以检验 TaCF 算法在较大数据量上的推荐性能。实验首先利用线性回归方法对三个训练集中的正负面评价时间效应曲线进行拟合，表3-9给出了这三个数据集上拟合结果的斜率和截距，表中结果显示在该组数据中正面评价的衰减速率都要大于负面评价的。然后利用公式 (3-15) 和公式 (3-16) 对用户的正负面评价进行衰减得到网络连边权重。表3-10给出了 TaCF 算法在三个数据集中的推荐结果。

为了方便比较，本章将 TaPNR 算法和 TaCF 算法在两组数据中相对其他算法的性能提升百分比进行比较。表3-11、表3-12和表3-13分别给出了在 MovieLens、Netflix 和 MovieTweetings 三个数据集上 TaPNR 和 TaCF 算法对比于 SNBI、UOS 和 IPF 算法的性能提升百分比结果。从结果可以发现，TaCF 推荐算法的推荐性能提升总体接近于 TaPNR 算法，表现出较为一致的趋势。在较稠密的 MovieLens

数据集上，相比于 SNBI、UOS 和 IPF 算法，不管是 TaPNR 算法还是 TaCF 算法都能够以牺牲少量个性化水平的代价显著的提高推荐结果准确性；而在较稀疏的 Netflix 和 MovieTweetings 数据集上，TaPNR 和 TaCF 算法除了显著提高推荐系统准确性以外，还能明显的提升推荐结果的新颖性和覆盖率，是准确性和个性化两类指标综合性能最好的两个算法。结果表明 TaCF 算法是一种可以代替 TaPNR 算法应用到真实推荐系统中的有效算法。

表 3-9 CorP 和 CorN 曲线拟合结果

| 数据集 | a_P | b_P | a_N | b_N |
|----------------|---------|--------|---------|--------|
| MovieLens | -0.0101 | 0.4578 | -0.0086 | 0.3712 |
| Netflix | -0.0115 | 0.5223 | -0.0104 | 0.4436 |
| MovieTweetings | -0.0095 | 0.2584 | -0.0052 | 0.2256 |

表 3-10 MovieLens、Netflix 和 MovieTweetings 数据集上 TaCF 算法的推荐性能

| 数据集 | F1 | 多样性 | 覆盖率 | 新颖性 |
|----------------|----------|----------|----------|-------------|
| MovieLens | 0.077092 | 0.807102 | 0.068428 | 1521.1383 |
| Netflix | 0.043614 | 0.835393 | 0.163379 | 2738.034665 |
| MovieTweetings | 0.053145 | 0.769488 | 0.189722 | 1058.740487 |

表 3-11 MovieLens 数据集上 TaPNR 和 TaCF 算法性能提升百分比对比结果

| 评价指标 | TaPNR | | | TaCF | | |
|---------|--------------|---------------|--------------|---------------|---------------|---------------|
| | SNBI | UOS | IPF | SNBI | UOS | IPF |
| F1(%) | 87.67 | 155.89 | 30.20 | 19.93 | 30.21 | 7.96 |
| 多样性 (%) | -8.76 | -12.45 | -4.53 | 13.32 | -8.68 | -0.54 |
| 新颖性 (%) | -4.31 | 2.64 | -2.1 | -17.87 | 7.53 | 45.71 |
| 覆盖率 (%) | -5.02 | -24.00 | -22.91 | 152.94 | -12.60 | -19.17 |

3.5 本章小结

本章首先分析了在电影评分数据集中用户对电影的正面评分和负面评价与用户将来兴趣的时序相关性。提出了一种基于正负面评价的时间感知推荐算法，该算法将用户正面和负面评分赋予了不同的时间衰减因子，同时算法还考虑了长短期用户偏好的不同比重。为了简化提出的推荐算法，提高计算效率，使模型能够有效的应用于真实在线系统中，本章利用线性回归的方式对正负面评价时间效应曲线进行了近似拟合，提出了基于曲线拟合的时间感知推荐算法。通过在三个真

表 3-12 Netflix 数据集上 TaPNR 和 TaCF 算法性能提升百分比对比结果

| 评价指标 | TaPNR | | | TaCF | | |
|---------|--------|--------|-------|----------------|--------------|----------------|
| | SNBI | UOS | IPF | SNBI | UOS | IPF |
| F1(%) | 19.65 | 29.60 | 21.33 | 50.14 | 41.04 | 62.22 |
| 多样性 (%) | 1.85 | -19.23 | -6.63 | 32.54 | -5.05 | 34.12 |
| 新颖性 (%) | -10.27 | 7.86 | -7.26 | -27.21 | 1.60 | -29.06 |
| 覆盖率 (%) | 63.64 | 41.54 | 13.15 | 1546.63 | 59.70 | 1471.40 |

表 3-13 MovieTweetings 数据集上 TaPNR 和 TaCF 算法性能提升百分比对比结果

| 评价指标 | TaPNR | | | TaCF | | |
|---------|---------------|--------------|---------------|--------------|---------------|--------------|
| | SNBI | UOS | IPF | SNBI | UOS | IPF |
| F1(%) | 19.31 | 76.53 | 10.19 | 41.60 | 76.25 | 24.78 |
| 多样性 (%) | 5.40 | -32.32 | 4.87 | 33.76 | -6.99 | 3.82 |
| 新颖性 (%) | -14.87 | 16.76 | -14.67 | -24.29 | 8.12 | -6.71 |
| 覆盖率 (%) | 147.41 | -27.00 | 200.00 | 87.63 | -12.32 | 46.89 |

实电影评分数据集上的实验表明，基于正负面评价的时间感知推荐算法能够比现有推荐算法提供更好的推荐准确性。此外，算法还能为稀疏数据集提供更好的个性化指标。通过对基于正负面评价的时间感知推荐算法和基于曲线拟合的时间感知推荐算法相较于基准算法的性能提升，发现基于曲线拟合的时间感知推荐算法虽然忽略了长短期兴趣的不同作用，但是该模型依然能够提供较高的推荐性能，具有较高的实际应用价值。

第四章 基于隐性反馈的时间感知推荐算法研究

4.1 问题的提出

评分数据能够使推荐系统更准确的衡量用户的偏好，但是要让用户主动为物品评分是一件很困难的事情，因为这会打扰用户的正常体验。在统计了系统中不同评分的数量分布后，研究人员发现用户通常只使用少数几个分数来表达他们的喜好，如 1 分和 5 分，其他的分数很少被用户使用。因此，为了简化评分系统，让更多的用户表达他们对物品的偏好，许多评分系统被改成了二级评分系统，即喜欢和不喜欢。虽然二级评分系统简化了用户的操作，但是用户使用率依然不高。相较于显性反馈，用户的隐性反馈更易于收集，例如用户浏览物品的时间、用户的浏览次数、地理位置等都可以通过后台系统悄悄地进行收集。

用户的购物和选择行为是一个动态过程。一方面，用户的兴趣会随着时间发生漂移，新的兴趣会不断出现，老的兴趣也会慢慢消失。另一方面，用户间的相互影响也存在着时间效应。例如，某用户会因为身边好友的影响开始接触二次元文化，购买好友推荐的商品。随着对二次元文化的了解，用户逐渐形成自己的兴趣，购物行为与好友逐渐发生差异，此时他的行为反而会开始影响他的好友。随着社交网站和评论系统的兴起，用户间的交流和相互影响越来越频繁。在线用户的行为与线下行为一样，一方面会受到其他用户的影响，另一方面也会影响其他人。在只有时间信息作为隐性反馈的系统中，用户之间的行为顺序可以看作是彼此相互影响的结果。有的用户习惯跟随主流，购买商品的时间经常在其他用户之后；有的用户更喜欢尝试新鲜的事物，经常在商品刚刚发布后就前去抢购。

除了用户个体表现出个性化的行为模式外，不同用户群体表现出的时间特性可能也是不尽相同的，用户与用户之间的影响关系也会随时间发生变化。如何从用户的隐性反馈行为中挖掘用户个体和用户群体的相关兴趣模式，并将其应用到推荐算法的研究中，是本章主要的研究内容。

4.2 用户隐性反馈行为的时间特性分析

4.2.1 用户个体行为的时间特性

随着移动互联网的快速发展，用户使用网络进行交流、消费、获取信息的频率越发的加快。用户在网络中的行为是动态变化的，表现出记忆性、周期性和偶然性等特点。为了分析用户在线行为中表现出的时间特性，本章对以下三种行为进行了相关研究：

(1) **用户个体的选择行为** (Similarity of individual user over time, 记为 SoT)。定义任意两个物品 o_α 和 o_β 之间的相似度为 $sim_{\alpha\beta} = \Gamma_\alpha \cdot \Gamma_\beta / |\Gamma_\alpha| \times |\Gamma_\beta|$, 其中, Γ_α 是物品 o_α 的 n 维向量形式。如果用户 u_i 购买了物品 o_α , 则 Γ_α 的第 i 个分量等于 1, 否则等于 0。为避免时间单位过于细化, 用户的选择行为被划分到不同的会话时间内。对于同一用户的任意两个会话, 以会话间每一对物品相似度的均值作为两个会话的相关性。针对用户 u_i , 可以得到该用户任意两个会话的时间间隔 Δt 与会话相关性之间的对应关系。将该用户相同时间间隔 Δt 对应的会话相关性取均值后, 可以得到该用户个体的一条关于时间间隔 Δt 与会话相关性的对应曲线。曲线上的值表示时间间隔为 Δt 时, 用户前后两个会话中选择的物品的相似度。将系统所有用户的时间间隔与会话相关性曲线组合在一起求均值后可以得到系统层面的用户个体 SoT 曲线。

(2) **用户整体的选择行为** (Similarity of all users in system, 记为 SoS)。将所有用户购买过的物品构成一个多重集合, 即如果一个物品被 n 位用户购买过, 则该物品在该集合内出现 n 次。随机选取多重集合内的两个物品计算相似度, 同分析用户个体的选择行为一样, 物品间的相似度仍使用原始物品向量间的余弦相似度。该过程重复进行 100000 次, 然后将所有结果的均值作为 SoS。该结果是将系统内所有用户作为一个用户个体看待, 忽略了用户个体之间的个性化差异, 但依然保留了用户的偏好信息, 即大众喜欢的流行物品被抽到的概率要大于小众物品。

(3) **系统随机的选择行为** (Similarity of items randomly, 记为 SoR)。将所有用户购买过的物品组成一个集合, 该集合是消除重复的, 即如果一个物品被购买过 n 次, 该物品在该集合中仅出现一次。对集合内的物品随机抽取两个计算其相似性, 该过程重复 100000 次, 全部结果的均值定义为 SoR。该结果彻底忽略了用户的偏好信息, 所有物品被等概率的抽出。

图4-1(a)、(b) 和 (c) 分别显示了在 Netflix、FriendFeed 和 MovieLens 数据集上 SoT、SoS 和 SoR 的实验结果。横坐标表示时间间隔, 纵坐标表示相隔 Δt 时间的两个会话中物品的平均相似度。从图中可以发现, 在三个数据集中 SoT 都是一条随着时间间隔的增加而逐渐降低的曲线。 $\Delta t = 0$ 的数值代表用户在每个自然月内所选择的物品间的平均相似度, 意味着用户的短期兴趣; 随着 Δt 的增加, SoT 曲线代表用户短期兴趣与长期兴趣间的差异性随时间的变化。图中 SoT 曲线的衰减趋势表明用户自身的兴趣长期来看发生了漂移, 随着时间间隔的增加, 相似度越来越低, 即短期和长期兴趣的差异性越来越大。而从短期来看, 用户的兴趣是比较稳定的, 较小的 Δt 对应的相似度值都比较高, 没有出现明显的抖动。此外, 在三个数据集上, SoT 的数值都要明显高于 SoS 和 SoR。这意味着相较于将所有

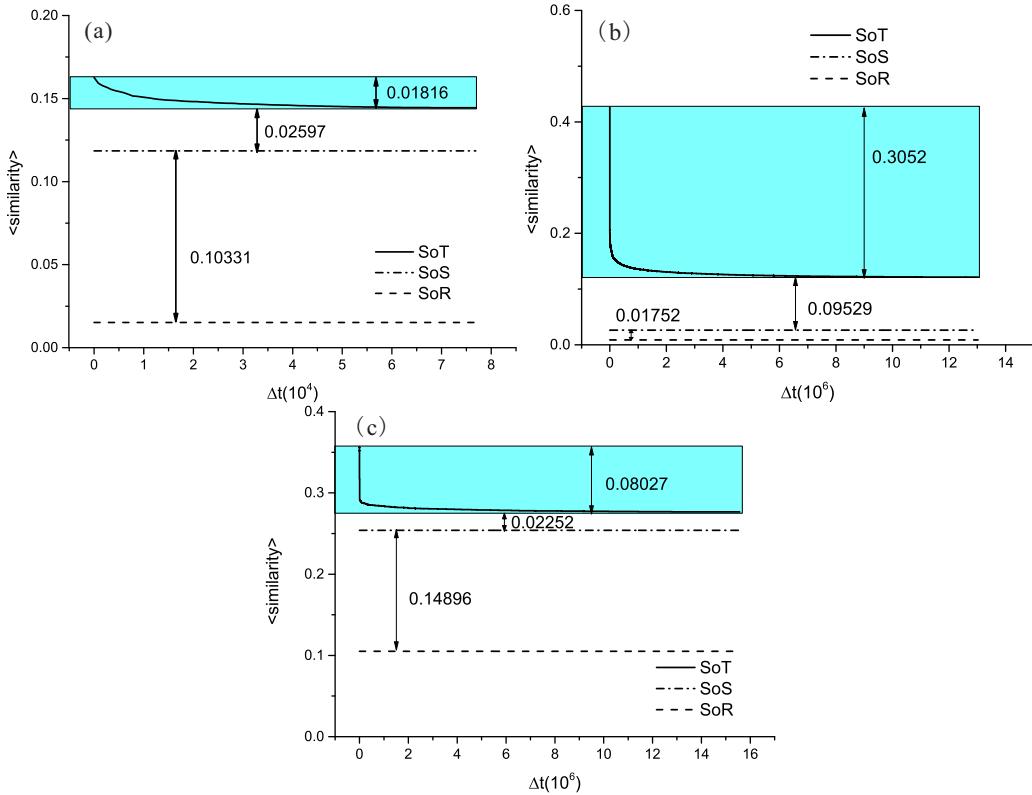


图 4-1 (a) Netflix、(b) FriendFeed 和 (c) MovieLens 数据集上 SoT、SoS 和 SoR 的统计结果

用户看作一个整体，每个个体用户的兴趣都比较集中，与其他用户相比具有更强的个性化，将用户作为整体看待会削减这种个性化程度。由于 FriendFeed 的数据类型不同于 Netflix 和 MovieLens，包含更多的主题，用户具有更多的选择，这导致 FriendFeed 网站中的物品生命周期普遍偏短。因此，从图4-1中可以发现，Netflix 和 MovieLens 的 SoT 最大值与最小值之间的差距要小于 FriendFeed 中的距离，兴趣漂移现象在 FriendFeed 中更加的严重。

4.2.2 引领型邻居和跟随型邻居

接着，为了分析用户之间的相互影响，本章为每个用户定义了两种邻居集合，分别称作引领型邻居和跟随型邻居。针对用户 u_i ，将与用户 u_i 购买过相同物品的用户作为候选邻居。对于共同购买过的物品 o_α ，如果候选邻居 u_j 购买该物品的时间早于目标用户 u_i ，则他是一个引领型邻居，否则是跟随型邻居。用户 u_i 购买物品 o_α 的这次行为可能是受引领型邻居的影响做出的，同时这次行为可能对跟随型邻居产生了后续影响。需要注意的是，对于共同购买过的另一个物品 o_β ，如果该候选邻居 u_j 购买物品 o_β 表现的时间顺序与物品 o_α 的相反，例如 $t_{i\alpha} < t_{j\alpha}$ 而

$t_{i\beta} > t_{j\beta}$ 时，用户 u_j 将既是目标用户 u_i 的引领型邻居，也是跟随型邻居。引领型和跟随型邻居的概念是针对用户的某次行为而定的，并不是用户和用户之间固定的关系，是一种基于行为时间感知的分类方法。本章依次根据系统中每个用户的每次行为 e_{ia} 将其他用户分为引领型邻居和跟随型邻居，并计算了两类邻居群体内部在时刻 t_{ia} 前购买物品的 SoS 值，然后将所有行为对应的引领型邻居的 SoS 和跟随型邻居的 SoS 的均值分别记为 SoG-L (Similarity of items with leading links) 和 SoG-F (Similarity of items with following links)。

表4-1显示了 Netflix、FriendFeed 和 MovieLens 三个数据集上 SoG-L 和 SoG-F 的实验统计结果。在 Netflix 和 FriendFeed 数据集中，SoG-F 的数值要高于 SoG-L，该结果表明跟随型用户购买和选择的物品彼此间具有更高的相似度，用户的兴趣更加集中。SoG-L 组别中的用户表现出更强的个性差异，他们主要根据自身的兴趣进行选择，而不是通过别人的建议来决定是否购买某商品。MovieLens 中的 SoG-F 数据要略低于 SoS，这与其他两个数据集中的情况相反。这可能是由于 MovieLens 数据集自身的特点导致，MovieLens 数据集去掉了度小于 20 的用户，使整个数据集较为稠密，用户与用户之间的行为重叠度更高，既是引领型又是跟随型邻居的用户占据较多的比例，无法从整体上将用户群体区分出来。

表 4-1 Netflix、FriendFeed 和 MovieLens 数据集上 SoG-L 和 SoG-F 统计结果

| 数据集 | V_{SoG-L} | V_{SoG-F} | $V_{SoG-L} - V_{SoS}$ | $V_{SoG-F} - V_{SoS}$ |
|-----------------------|-------------|-------------|-----------------------|-----------------------|
| Netflix (10^4) | 0.12186 | 0.12789 | 0.00336 | 0.00939 |
| FriendFeed (10^6) | 0.08285 | 0.08987 | 0.05593 | 0.06328 |
| MovieLens (10^6) | 0.25873 | 0.25382 | 0.00460 | -0.00031 |

通过实验分析表明，用户的长期偏好和短期偏好存在着较大的差异性，在设计推荐系统时需要考虑用户不同时期的兴趣偏好。同时，引领型用户和跟随型用户对网络中其他用户的行为敏感程度是不同的，跟随型用户较容易受他人影响而做出购物决定。因此在设计推荐算法时需要将这些因素综合考虑，以准确捕捉用户行为的时序特性。

4.3 基于资源分配的时间感知推荐模型

Zhou 等人^[77] 根据物质扩散的思想，提出了一种基于用户-物品二部图的资源分配算法用来向用户推荐商品。算法将用户对物品的偏好表示为网络中节点拥有的资源，根据目标用户的购买行为为物品节点赋予初始资源，如果用户购买过该物品则赋值为 1，否则为 0。偏好可以在网络中进行扩散，扩散过程类似于随机游

走，每次以度分之一的概率传递到下一个节点。整个过程分为两个阶段：传播阶段和重新分配阶段。在传播阶段，资源从物品节点传播到用户节点；在重新分配阶段，资源从用户节点重新分配到物品节点。本章在这两个阶段中分别考虑了用户之间的时序影响和用户个体的兴趣漂移现象，提出了基于资源分配的时间感知推荐算法（Time-aware Diffusion-Based Algorithm，记为 TaDb）。

4.3.1 用户之间的时序影响

资源首先从目标用户购买过的物品开始扩散到用户节点。在这个过程中，资源的转移其实是在寻找跟目标用户购买过相同物品的其他用户，这些用户可以称为目标用户的邻居。邻居节点获得到的资源代表他与目标用户节点的相似性，对于目标用户 u_i ，用户节点 u_j 得到的资源为：

$$f_j = \sum_{\alpha=1}^n \frac{a_{j\alpha}a_{i\alpha}}{k_\alpha} \quad (4-1)$$

其中， k_α 表示物品 o_α 的度， $a_{j\alpha}$ 表示用户 u_j 是否购买过物品 o_α 。从公式 (4-1) 可以发现，邻居与目标用户的相似关系取决于共同购买的物品的流行度，如果该物品被很多人购买过，那么衡量用户与目标用户之间相似性的权重就会变小。如果两个用户共同购买了很多冷门物品，那么他们的相似度就会很高。在这个过程中，原始方法并没有考虑两个用户之间共同购买物品的先后顺序。通过分析网络中引领型用户和跟随型用户的行为特性，可以发现两类用户选择物品的行为特性存在一定的差异，对其他用户的影响也不同。因此，本章首先考虑用户共同购买物品的时序影响。

给定目标用户 u_i ，依次将目标用户与其购买的物品间形成的连边作为参考对象。例如，用户 u_i 购买了物品 o_α ，以连边 $e_{i\alpha}$ 作为基准。考虑其他同样购买了物品 o_α 的用户，如果用户 u_j 购买物品 o_α 的时间 $t_{j\alpha}$ 早于用户 u_i 购买物品 o_α 的时间 $t_{i\alpha}$ （即 $t_{j\alpha} < t_{i\alpha}$ ），则将连边 $e_{j\alpha}$ 作为一条 Leading 连边；相反，如果 $t_{j\alpha} > t_{i\alpha}$ ，则连边 $e_{j\alpha}$ 作为一条 Following 连边。Leading 连边和 Following 连边所对应的用户被分别分配到集合 $Leader(i)$ 和 $Follower(i)$ 中，其中 $Leader(i)$ 代表根据用户 u_i 的行为确定的引领型邻居用户集合， $Follower(i)$ 表示根据用户 u_i 的行为确定的跟随型邻居用户集合。图4-2给出了一个样例，图中用户 u_1 是目标用户，当以连边 e_{11} 作为参考对象考虑资源从物品节点 o_1 扩散到用户节点时，因为 $t_{11} < t_{21}$ ，连边 e_{21} 是一条 Following 连边，用户 u_2 归类到跟随型邻居中；而 $t_{11} > t_{31}$ ， e_{31} 是一条 Leading 连边，用户 u_3 则归类为引领型邻居。类似的，当以连边 e_{12} 为参考对象时，用户 u_2 被归类为引领型邻居，连边 e_{22} 是一条 Leading 连边。

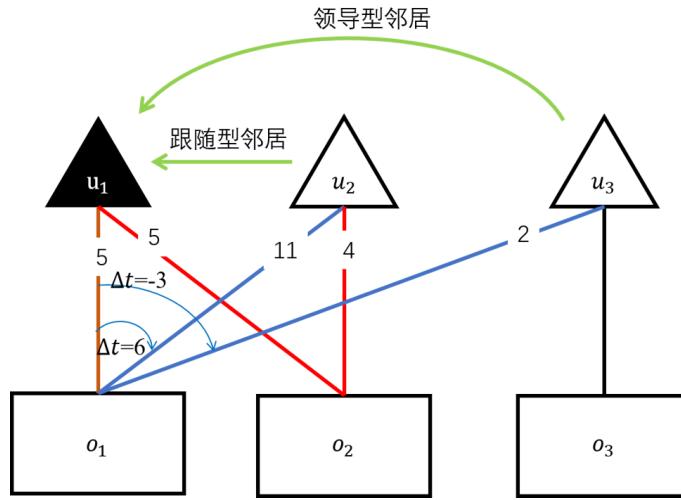


图 4-2 基于资源分配的时间感知推荐算法示意图

当资源从物品节点传递到用户节点时，根据不同类型的连边（Leading 或者 Following）对资源进行不同速率的衰减，连边权重定义为：

$$w_{\beta j} = \begin{cases} \rho \times e^{-\frac{1}{a}|t_{j\beta} - t_{i\beta}|}, & \text{如果 } t_{j\beta} < t_{i\beta} \\ (1 - \rho) \times e^{-\frac{1}{b}|t_{j\beta} - t_{i\beta}|}, & \text{如果 } t_{j\beta} > t_{i\beta} \end{cases} \quad (4-2)$$

其中， $a = \sum_{\alpha \in Leader(i) \cap O(j)} (t_{i\alpha} - t_{j\alpha})$, $b = \sum_{\alpha \in Follower(i) \cap O(j)} (t_{j\alpha} - t_{i\alpha})$, $O(j)$ 表示用户 u_j 购买过的物品集合，调节参数 $\rho \in [0, 1]$ 表示引领型用户所占的权重。因为在真实世界中，每个用户都有自己的时间敏感尺度，使用一个绝对的衰减速率作为基准显得不合适。本算法计算了用户 u_i 作为引领型用户时，与用户 u_j 共同购买过的物品对应的总时间间隔 a ，以及用户 u_j 作为引领型用户时，与用户 u_i 共同购买过的物品对应的总时间间隔 b ，并以 $1/a$ 和 $1/b$ 作为时间衰减速率来代表用户之间非对称的相互影响。

从物品节点到用户节点的连边权重是基于目标用户和该物品间交互行为的时间信息作为参考的，根据行为的发生顺序给予不同的权重，这是一种基于行为时间感知的赋权过程。

4.3.2 用户个体的兴趣漂移

当偏好从初始物品节点扩散到邻居用户节点以后，会接着从邻居用户节点继续传递回物品节点。在这个过程中，资源的转移是在通过聚合相似用户的共同兴趣来预测目标用户对特定物品的喜爱概率。资源经过两步扩散以后，物品节点 o_α

最终获得的资源可以用以下公式表示：

$$f_\alpha' = \sum_{l=1}^m \frac{a_{l\alpha} f_l}{k_l} \quad (4-3)$$

其中， k_l 表示用户 u_l 的度。如果一个用户购买过太多物品，那么他购买的特定物品对于表征用户兴趣的能力就相应的很小。通过实证分析可以发现用户的兴趣在长期来看是存在漂移的，近期的购物选择对下个时刻的购买决定具有更大的影响，因此用户偏好的贡献度在随着时间推移时也需要进行一定程度的衰减。在资源从用户节点传递到物品节点的过程中，算法考虑了个体用户的兴趣漂移影响。

在资源从邻居用户节点传递到物品节点时，根据邻居用户选择该物品的时间和当前时间对资源进行衰减，以强调邻居用户近期行为的重要性。衰减权重定义为：

$$w_{j,\beta} = e^{-\frac{1}{\tau_j} |T - t_{j\beta}|} \quad (4-4)$$

其中， $\tau_{ji} = \sum_{\alpha \in Leader(i) \cap O(j)} (t_{i\alpha} - t_{j\alpha}) / |Leader(i)|$ ，表示用户 u_i 过去跟随用户 u_j 的平均滞后时间， T 表示进行预测时的系统当前时间。例如，在图4-2中，用户 u_2 购买物品 o_1 和 o_2 的时间分别是 11 和 4，可以认为最近购买的物品 o_2 更能表现用户 u_2 近期的兴趣爱好，通过时间衰减后将得到更大的推荐概率。

通过将用户-物品二部图中的连边考虑成有向边后，利用公式 (4-2) 和 (4-4) 对不同类型的连边赋予不同的权重。针对目标用户 u_i ，物品 o_α 最终获得的资源数等于：

$$f_\alpha' = \sum_{l=1}^m \frac{a_{l\alpha} w_{l,\alpha}}{k_l} \sum_{\beta=1}^n \frac{w_{\beta,l} a_{l\beta} f_\beta}{k_\beta} \quad (4-5)$$

每个物品最终获得的资源数量可以表示用户对物品喜欢的概率，也可以作为目标用户对该物品的一种预测评分。系统将目标用户未购买过的物品按照最后得到的资源进行降序排列，排名靠前的 L 个物品被组成推荐列表推荐给目标用户。

4.4 算法性能评测

4.4.1 实验数据集和对比算法

本章采用了 Netflix、MovieLens、FriendFeed 和 Delicious 四个数据集。Netflix 数据集^①是对原始数据集的一个随机采样，首先从在 2005 年 1 月到 2005 年 5 月的记录里随机抽取 5000 名用户作为观察目标，然后将这些用户在这段时间内的所有评分记录作为实验数据。FriendFeed 数据集是对从 2010 年 1 月到 2010 年 8 月间的

^① 可从 https://github.com/wjleeUESTC/thesis_data/tree/master/data4 下载

用户偏好数据进行随机采样后形成的数据集。MovieLens 采用了 MovieLens 100k 的数据集，从中去除了只在 1 个自然月中有数据的用户。表4-2展示了三个数据集的基本统计信息。

表 4-2 数据集基本统计信息

| 数据集 | 用户数 | 物品数 | 稀疏性 | 时间跨度 |
|------------|-------|-------|-------|-------|
| Netflix | 5000 | 12493 | 0.56% | 5 个月 |
| MovieLens | 835 | 1630 | 6.13% | 6 个月 |
| FriendFeed | 10967 | 8020 | 0.12% | 7 个月 |
| Delicious | 1541 | 47484 | 0.10% | 10 个月 |

在划分训练集的时候，将最后一个自然月中包含的数据作为测试集，而发生在这之前的剩余数据作为训练集，然后从测试集中剔除了未在训练集中包含的用户，以此模拟真实系统环境下严格利用历史数据预测未来兴趣的场景。在所有数据集中，以训练集中最大的时刻作为系统的预测时间，并为每位用户推荐 10 个商品。

为了验证本章提出的基于资源分配的时间感知推荐算法的性能，本章将该算法与标准的资源分配算法（NBI）^[77]、基于资源分配的时间加权算法（TWN）^[40]以及基于用户的协同过滤算法（UCF）^[155]进行了对比。其中，标准资源分配算法和基于用户的协同过滤算法是时间无关的静态推荐算法，而基于资源分配的时间加权算法是与时间信息相关的算法。基于资源分配的时间加权算法一方面通过时间衰减方法对最近发生的行为给予更大的权重，另一方面对同一用户产生的两次行为进行了比较，如果两次行为时间比较接近，则将资源在这两次行为间的传递赋予更大的权重。基于用户的协同过滤算法中为每名用户选取了 100 名最相似的用户作为邻居。

4.4.2 时间感知算法的性能分析

图4-3显示了不同推荐算法在 Netflix、MovieLens、FriendFeed 和 Delicious 四个数据集上推荐准确率随算法参数的变化情况，横坐标表示算法参数，纵坐标表示推荐列表长度等于 10 时的推荐准确率。TaDb 算法中的参数 $\rho \in [0, 1]$ 表示引领型用户所占的比重。TWN 算法中的参数 λ 表示时间衰减的速率，NBI 和 CF 算法不包含参数，因此在图上这两个算法的结果是两条水平线。

TWN 算法主要考虑的是同一用户的两次行为之间的时间效应，认为相隔越近的两次行为具有越高的相关性。算法使用参数 λ 对两次行为的时间间隔进行权重衰减， λ 越大对时间作用衰减的越厉害。从图4-3可以发现，在 Netflix 和 MovieLens

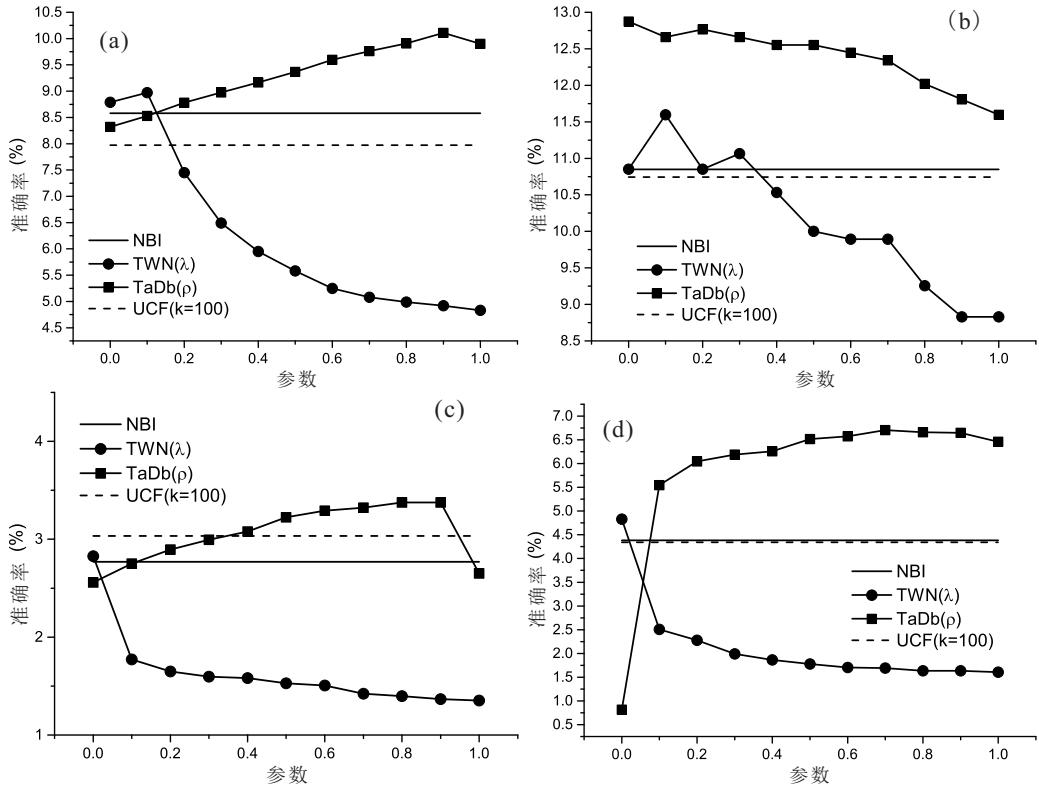


图 4-3 不同推荐算法随参数变化的推荐准确性情况。(a) Netflix; (b) MovieLens; (c) FriendFeed; (d) Delicious

数据集中，最优准确率对应的 λ 都在 0.1 附近，说明用户个体行为的时间作用并不需要进行过分的衰减，个体用户行为间的差异性并不显著。此外，在 FriendFeed 和 Delicious 这种比较稀疏的数据集中，最优准确率对应的 λ 值都在 0 附近，说明在这两个系统中用户个体的行为之间并不需要考虑时间的差距，用户个体行为的时间特性对提高推荐系统性能起到的作用并不突出。

表 4-3 时间因素对 TaDb 算法推荐性能的影响

| 数据集 | 考虑时间信息 (%) | 不考虑时间信息 (%) |
|------------|--------------|-------------|
| Netflix | 9.32 | 8.69 |
| MovieLens | 12.61 | 10.86 |
| FriendFeed | 3.23 | 2.76 |
| Delicious | 6.56 | 5.22 |

TaDb 算法主要考虑的是用户之间的行为时间效应，时间衰减的速率直接通过用户的历史数据进行推断。为了分析 TaDb 算法中时间特性的作用，将公式 (4-2) 中的参数 a 和 b 设置为无穷大，以忽略用户间的时间特性。为了避免其他参数的干扰，将参数 ρ 设置为 0.5。表 4-3 给出了考虑和不考虑用户间行为的时间因素时推荐

准确率的对比结果。从表中可以发现，不同用户行为间的时间间隔信息对提高推荐结果的准确性具有明显的积极作用，Netflix、FrienFeed、Delicious 和 MovieLens 数据集上的准确率都要比不考虑时间因素时高出 1%。用户间相互影响的时间特性要比用户个体的时间特性更为普遍。

4.4.3 不同算法的实验结果对比

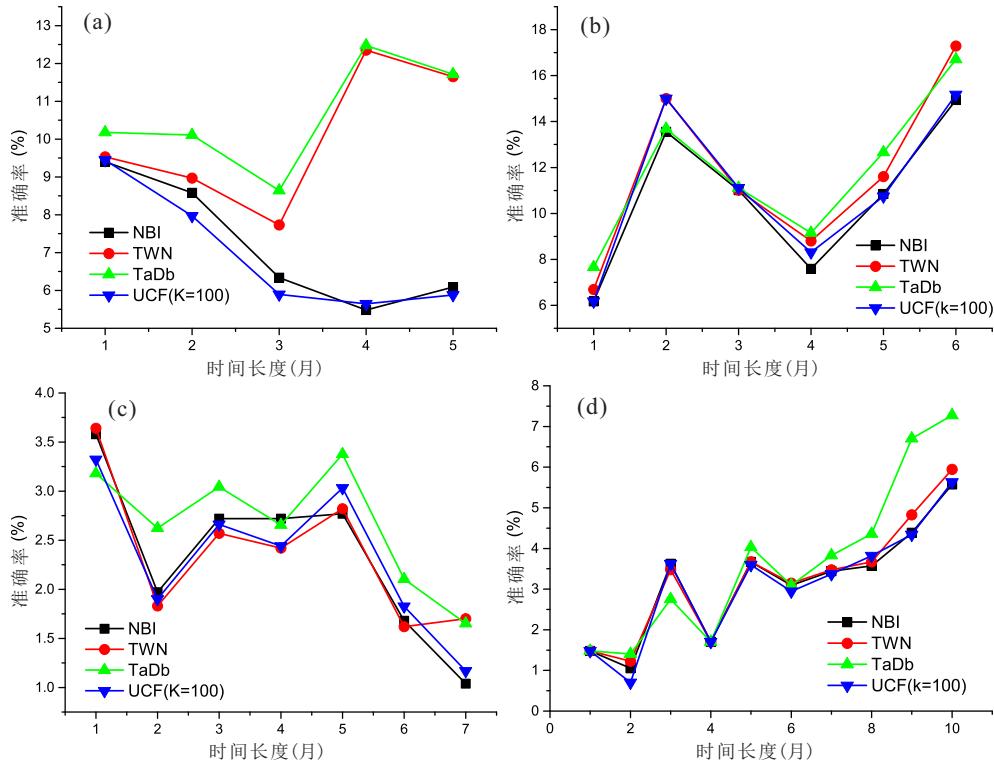


图 4-4 不同算法的推荐准确率与训练集时间长度的关系。(a) Netflix; (b) MovieLens;
(c) FriendFeed; (d) Delicious

图4-4显示了 NBI、UCF、TWN、TaDb 算法在 Netflix、MovieLens、FriendFeed 和 Delicious 数据集上的推荐准确率以及他们随训练集时间长度的变化情况，横坐标是以月为单位的时间长度，纵坐标代表推荐准确率。从结果中可以发现，大多数情况下，本章提出的 TaDb 算法（绿色）的推荐准确率都要高于其他三个算法。在 Netflix 数据集上，当训练集只有 1 个月的数据量时，TaDb 算法依然能够提供很好的推荐性能。而在 FriendFeed 和 Delicious 这类稀疏数据集上，TaDb 算法更适合在数据量比较充分的时候发挥效果，较为稀疏的数据中无法捕获足够的用户之间的关系。之前在 MovieLens 数据集中观察到 SoG-F 要小于 SoS，该数据集中用户的跟随效应并不明显，这也导致了 TaDb 算法在 MovieLens 数据集上并没有比 TWN 算法有明显的准确率提升。

为了对比不同算法的综合表现，本章将每个算法在不同时间长度的训练集上得到的结果计算了均值，表4-4给出了求均值后 TaDb 算法相较于其他三个对比算法准确率提高的百分比结果。从表中可以看出，时间相关的推荐算法的准确率要明显高于另外两个非时间相关的静态推荐算法。而在时间相关的推荐算法中，本章提出的 TaDb 算法在四个数据集中都要优于 TWN 算法，在 Netflix、MovieLens、FriendFeed 和 Delicious 数据集上分别提高了 6.58%、2.74%、15.09% 和 9.20%。由此可以看出，通过时间信息反映的用户购物行为的先后顺序对提高推荐系统性能起着积极的作用。

表 4-4 TaDb 相比于其他算法在准确率方面提高的百分比

| 数据集 | NBI (%) | TWN (%) | UCF (%) |
|------------|---------|---------|---------|
| Netflix | 56.50 | 6.58 | 6.03 |
| MovieLens | 12.53 | 2.74 | 8.90 |
| FriendFeed | 19.64 | 15.09 | 17.80 |
| Delicious | 13.59 | 9.20 | 20.47 |

推荐系统的推荐准确率是性能评测的核心，但是盲目崇拜准确性指标会伤害推荐系统，会使得用户视野变得越来越狭窄。在追求准确率的同时，也需要关注推荐系统的个性化水平。个性化水平包括被推荐物品的多样性、新颖性和覆盖率等方面。接下来，选取 MovieLens 和 Netflix 两个相对稠密的数据集，将 TaDb 与 NBI 和 TWN 这两个基于网络结构的推荐算法就个性化水平进行分析。

表 4-5 对比算法的推荐多样性比较

| 数据集 | NBI | TWN | TaDb |
|-----------|--------|---------------|--------|
| Netflix | 0.6108 | 0.7499 | 0.5299 |
| MovieLens | 0.7938 | 0.8102 | 0.8072 |

表4-5给出了三种基于资源分配的推荐模型的多样性结果。结果表明基于用户个体行为时间特性的 TWN 算法可以更好的把握个体用户兴趣漂移情况，进一步提高推荐结果的多样性。而基于用户间行为时间特性的 TaDb 算法更关注于用户的群体效应，在显著提升推荐准确性的同时牺牲了多样性指标。当想要综合考虑推荐准确率和多样性指标时，可以调节参数 ρ 来找到满足需求的平衡点。

表4-6给出了三种基于资源分配的推荐模型的新颖性结果。从表中可以发现，TaDb 提供的推荐结果新颖性要略好于 TWN 和 NBI 算法，能够更好的发掘用户兴趣中的小众商品，对解决长尾商品的销售问题有着重要的价值。

表 4-6 对比算法的推荐新颖性比较

| 数据集 | NBI | TWN | TaDb |
|-----------|---------|---------|---------------|
| Netflix | 2469.56 | 1057.64 | 965.20 |
| MovieLens | 321.28 | 76.44 | 60.27 |

4.5 本章小结

本章的研究动机是用户的在线购物行为会受到其他用户的影响并且继而影响其他人，即使他们并非朋友关系。本章首先分析了用户个体行为、用户整体行为和系统随机性行为之间的差异，实证结果表明：1) 用户的购买行为并不是完全的随机行为，具有很强的个性化水平；2) 用户的短期兴趣和长期兴趣之间存在差异，用户在较近的会话中表现出的短期兴趣较为稳定，而随着时间的推移，用户的兴趣表现出漂移的现象。接着，基于用户行为的时间信息，本章将用户的邻居分为了引领型和跟随型两个群体，研究了群体内部选择物品的相似度情况。实验结果表明引领型用户倾向依靠自己的兴趣购买商品，彼此之间表现出明显的多样性，而跟随型用户购买的物品相似度较高，通常是受到其他用户的影响从而决定对某物品的购买行为。基于实证结果，本章提出了一种基于资源分配的时间感知推荐算法 TaDb。该算法在两阶段资源传递过程中考虑了用户间的相互影响和个体的兴趣漂移因素。通过在四个数据集上与三个基准算法的对比，实验结果表明 TaDb 算法相较于静态算法可以显著提高推荐准确率，而在用户交互现象较明显的数据集中，TaDb 算法的推荐准确性要优于基于用户个体行为时间特征的推荐算法。

第五章 基于物品近期流行度的推荐算法研究

5.1 问题的提出

近期，诸如资源分配和热传导等物理动力学思想被研究人员加入到了推荐算法的设计中。无论是基于资源分配还是基于热传导的算法，都是依赖于用户-物品二部图形式的网络结构，当偏好在网络中传递时，资源或热量的分配取决于用户或者物品的度信息。其中，物品的度指物品被多少用户购买过，即物品的长期流行度。在产品发布之前，仅根据产品特征来预测用户对该产品感兴趣程度，被证明是十分困难的任务^[156]，但是使用二部图这种网络结构来预测最近一段时间内可能会流行的物品被证明是可行的^[157]。物品的近期流行度是预测物品未来流行趋势的一个重要的特征，它可以很好的反映网络增长的概率^[158-160]。预测哪些物品会变得流行对供应商和它们的市场策略有着十分重要的意义。例如，如果视频网站知道哪一部视频会在近期变得火爆，那么就可以提前将该部电影备份到多个下载节点中保证用户观看视频的流畅性，以此增加用户体验。

物品的长期流行度和近期流行度是系统从物品角度观察到的时间特性。物品从添加、更新，最后到淘汰的整个生命周期从侧面反应了用户偏好的变化。如何从物品的角度捕捉系统中用户的行为时序特性，将物品的近期流行度应用到推荐算法中以提高推荐性能，这是本章主要的主要研究内容。

5.2 物品的时间特性研究

物品的长期流行度代表物品的受欢迎程度，长期流行度越高代表被越多的用户所喜爱。本章首先研究系统中物品的长期流行度分布情况。MovieLens 和 Netflix 数据集是用来研究推荐算法最常使用的两个数据集。图5-1显示了 MovieLens 数据集中物品的度（即物品的长期流行度）分布，可以很清楚的发现该分布呈现幂率特性。数据集中只有很少的一部分商品极度流行，而绝大部分商品仅被很少的用户评过分。图5-2给出了 MovieLens 和 Netflix 数据集中物品的长期流行度与平均在线天数之间的关系。从图中可以看到物品的长期流行度与平均在线天数是成正比的，几乎不会出现很热门的物品只流行几天的情况。

物品最近流行度在趋势预测领域扮演着重要的角色^[157]。在推荐系统中，如果只考虑物品的长期流行度，直观的结果是会导致很多已经过时的大度商品被推荐给用户。如图5-3所示，该图展示了在 MovieLens 数据集中“Pink Floyd: The Wall”（红色）和“Insomnia”（黑色）两部电影从 1996 年 10 月开始每个月的评分数量随

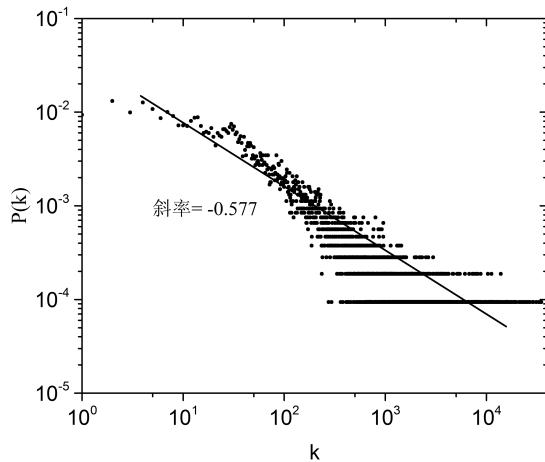


图 5-1 MovieLens 数据集中物品流行度的分布情况

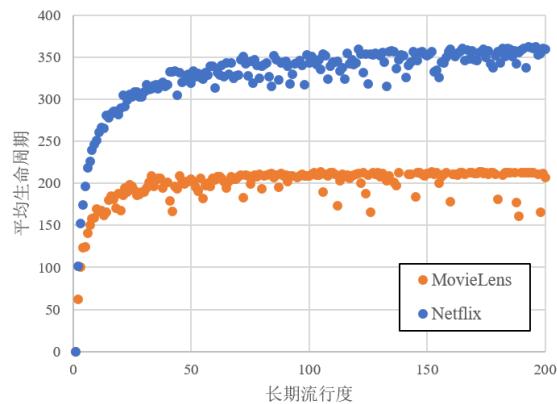


图 5-2 物品长期流行度与平均在线天数之间的关系

时间的变化情况。“Pink Floyd: The Wall”发布于 1982 年，而“Insomnia”发布于 2002 年。虽然在数据集中对这两部电影评分的用户数都是 3245，即物品的总流行度是一样的，但是从图中可以发现两者的每月评分数量趋势曲线有着较大差异。红色曲线代表的电影在第 36 个月至第 51 个月中的流行度出现了爆发性的增长，而在第 53 个月后每月评分人数就一直处于 40 以下（除了第 104 个月的峰值，该峰值可能是由于 2005 年 3 月 MovieLens 网站的论坛系统上线，致使大量用户涌入系统，增加了电影评分数数据，系统中每部电影在该时间段内的评分数都有明显的上升）。黑色代表的电影在 2002 年 5 月发布后流行度就一直高于红色电影，系统在此期间推荐“Insomnia”可能会被更多的观众采纳。直到 2007 年 7 月后，两者的趋势变得较为相似，此时为刚看过“Pink Floyd: The Wall”或“Insomnia”的用户推荐另一部电影可能会显得更有价值。

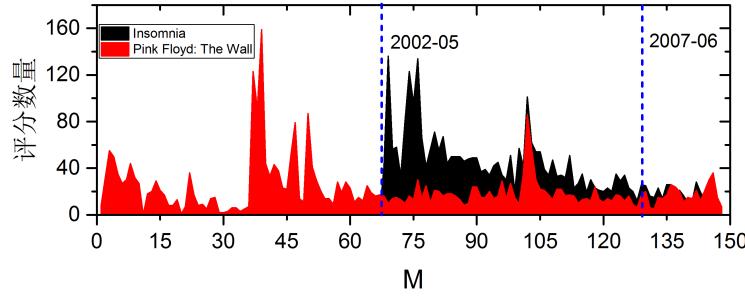


图 5-3 “Pink Floyd: The Wall” 和 “Insomnia” 评分人数随时间变化情况

5.3 基于近期流行度的推荐算法

用户-物品二部图 $G(U, O, E)$ 可以用一个邻接矩阵 $A = \{a_{ia}\}$ 表示，如果用户 u_i 选择过物品 o_a （即 u_i 与 o_a 存在连边）则 $a_{ia} = 1$ ，否则 $a_{ia} = 0$ 。给定任意一个时刻 t ，仅考虑发生在该时刻前的数据记录可以得到一个邻接矩阵 A 的时间切片，标记为 $A(t)$ 。在时刻 t 时，用户选择过的物品数量（即用户的度）和物品被选择过的数量（即物品的度）分别用 $k_i(t) = \sum_a a_{ia}(t)$ 和 $k_a(t) = \sum_i a_{ia}(t)$ 表示。

5.3.1 物品的近期流行度

给定当前时间 t^* 及一个表示从过去到当前时刻的时间窗口长度 T_p ，物品 o_a 的近期流行度被定义为从 $t^* - T_p$ 时刻开始到 t^* 时刻为止物品 o_a 的度的变化情况，公式表示为：

$$\Delta k_a(t^*, T_p) = k_a(t^*) - k_a(t^* - T_p) \quad (5-1)$$

如果给定一组合适的时间窗口长度 T_p ，近期流行度可以用来衡量用户对物品 o_a 兴趣的动态演化过程。此外，为了避免混淆，在本章余下部分，当讨论到流行度时主要指物品的总流行度或长期流行度。

为了分析物品近期流行度对用户将来兴趣的影响，本章从 Netflix 数据集中随机选择了 6000 个物品作为观察对象，图 5-4 给出了这些物品在 $(t^* - T_p, t^*]$ 时间段内的度（物品购买人数）与在 $(t^*, t^* + T_F]$ 时间段内的度的对应关系，其中 T_p 设置为 10 天， T_F 设置为 30 天。可以发现两个时间段内的度呈现显著的正相关特性，其 Pearson 相关系数等于 0.95442。图 5-5 显示了 Pearson 相关系数与时间窗口长度 T_p 之间的关系。随着 T_p 的变化，Pearson 数值变化的十分明显。当选取的窗口大小等于 10 天时相关性变得最大，太小的时间窗口会使最近流行度包含的可用信息过少而不能进行准确的推断，而太大的时间窗口会包含较多的历史信息而将已经过气的商品的权重变大导致相关性变低。通常情况下，在过去一段时间 $(t^* - T_p, t^*]$

内十分流行的物品很大概率会在将来的 $(t^*, t^* + T_F]$ 时间内继续维持一定的流行度。当只考虑全局总流行度时，一些过去流行过但近期不再被关注的物品会以较大的概率推荐给用户，这些推荐信息会影响到推荐系统的整体准确度。

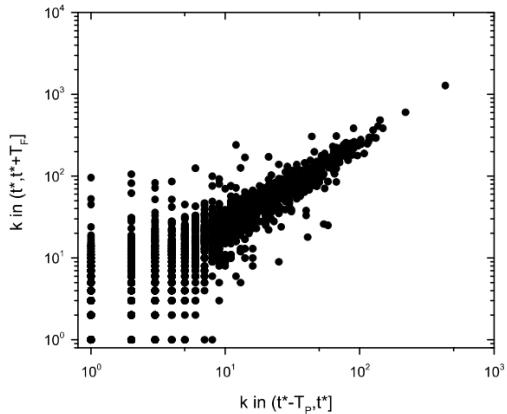


图 5-4 $(t^* - T_p, t^*]$ 和 $(t^*, t^* + T_F]$ 间物品的度相关性

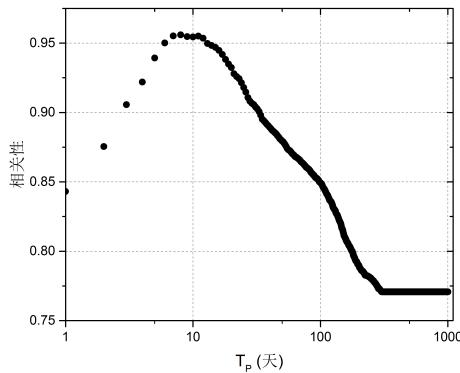


图 5-5 皮尔逊相关性与时间窗口 T_p 的关系

5.3.2 融合近期流行度的推荐算法

资源分配算法 (Mass Diffusion) 和热传导算法 (Heat Conduction) 是利用物理动力学概念解决推荐算法的两个基础算法。基于资源分配的推荐算法将推荐问题定义成以资源表示的用户偏好在网络结构中的概率转移问题。根据目标用户历史购买行为，将已经购买的物品的资源赋值为 1，未购买的赋值为 0。资源首先从物品节点传递给用户节点，然后从用户节点传递回物品节点。传递过程中一个节点的资源被平均分配给它的邻居节点。转移矩阵 W 中的元素 $w_{\alpha\beta}$ 表示物品节点 o_β 传

递给物品节点 o_α 的能量, 定义为:

$$w_{\alpha\beta}^{MD} = \frac{1}{k_\beta} \sum_{i=1}^m \frac{a_{ia}a_{i\beta}}{k_i} \quad (5-2)$$

基于热传导的推荐算法的传递过程与资源分配算法类似, 主要区别在于偏好传递过程中节点获得的能量平均的来自于邻居节点, 元素 $w_{\alpha\beta}$ 定义为:

$$w_{\alpha\beta}^{HC} = \frac{1}{k_\alpha} \sum_{i=1}^m \frac{a_{ia}a_{i\beta}}{k_i} \quad (5-3)$$

为了研究物品近期流行度在推荐算法中的作用, 本章引入调节参数 $\gamma \in [-1, 1]$ 将物品的长期流行度和近期流行度进行了融合, 公式定义如下:

$$s_\alpha(t^*) = k_\alpha(t^*) + \gamma \Delta k_\alpha(t^*, T_P) \quad (5-4)$$

基于公式 (5-4) 对两种流行度的融合, 本章提出了融合近期流行度的资源分配算法和融合近期流行度的热传导算法, 分别记为 HP-MD 和 HP-HC。HP-MD 和 HP-HC 算法的转移概率矩阵 W 的元素 $w_{\alpha\beta}$ 分别定义为:

$$w_{\alpha\beta}^{HP-MD}(t^*) = \frac{1}{k_\beta(t^*) + \gamma \Delta k_\beta(t^*, T_P) + \epsilon} \sum_{i=1}^m \frac{a_{ia}(t^*)a_{i\beta}(t^*)}{k_i(t^*)} \quad (5-5)$$

$$w_{\alpha\beta}^{HP-HC}(t^*) = \frac{1}{k_\alpha(t^*) + \gamma \Delta k_\alpha(t^*, T_P) + \epsilon} \sum_{i=1}^m \frac{a_{ia}(t^*)a_{i\beta}(t^*)}{k_i(t^*)} \quad (5-6)$$

其中, 当 $\gamma \in [-1, 0)$ 时, 近期流行的物品将得到较多的资源并被放到推荐列表排名靠前的位置; 当 $\gamma \in (0, 1]$ 时, 这些物品将会排在推荐列表尾部; 当 $\gamma = 0$ 时, 就退化为标准的资源分配和热传导算法。参数 ϵ 的作用是避免参数 $\gamma = -1$ 时分母等于 0, 本章将 ϵ 设置为 10^{-4} 。

5.3.3 融合近期流行度的推荐算法性能分析

本章使用了 MovieLens 和 Netflix 两个电影评分数据集来分析最近流行度在推荐算法中的作用。MovieLens 采用了最小的 MovieLens 100k 的数据集。本章主要关注的问题是推荐系统中的 Top-N 问题, 因此仅保留了原始数据集中评分高于 2 分的数据记录来表征用户的偏好, 在粗粒化处理后剩余 82520 条记录。Netflix 数据集^①是在原始数据集 2005 年 1 月至 2005 年 12 月期间随机抽取的子数据集, 包

^① 可从 https://github.com/wjleeUESTC/thesis_data/tree/master/data5 下载

含 10000 名用户和 6000 部电影，采用同样的粗粒化处理后共包含 535069 条记录。表5-1显示了本章实验数据集的基本统计信息。

表 5-1 数据集基本统计信息

| 数据集 | 用户数 | 物品数 | 记录数 | 稀疏度 |
|-----------|-------|------|--------|-----------------------|
| MovieLens | 943 | 1687 | 100000 | 6.30×10^{-2} |
| Netflix | 10000 | 6000 | 535069 | 9.02×10^{-2} |

数据集中所有评分记录按照时间顺序进行排列。给定一个时间窗口长度 T_P ，定义发生在随后一个时间窗口长度 T_P 内的记录作为测试集，剩余的记录作为训练集。用户的电影观看行为往往具有显著的月份效应。例如在中国，由于六一儿童节的原因，人们倾向在六月带着小孩去观看卡通类电影；而在二月，因为情人节的关系，人们喜欢带着伴侣观看浪漫电影。本章的主要目标是预测在近期会抓住用户眼球的电影。因此，将 T_F 设置为 30 天，即预测下一个月内用户会选择的电影。经过对 T_F 的设置，MovieLens 数据集上 1998 年 4 月的数据和 Netflix 数据集上 2005 年 12 月的数据被作为本次实验的测试集（由于原始数据的限制，MovieLens 数据集中 1994 年 4 月份的数据只包含 22 天的记录）。

图5-6给出了在 MovieLens 数据集上 HP-MD 和 HP-HC 算法的推荐性能随参数 γ 变化的情况，包括 F1 指标、多样性和新颖性三个维度。图中左侧纵坐标表示 HP-MD 算法对应的结果（蓝色曲线），右侧纵坐标表示 HP-HC 算法对应的结果（黑色曲线）。从图中可以发现，当参数 γ 从 -1 变化到 1 时，HP-MD 算法的 F1 指标变化的很小，最低点和最高点仅相差 0.0038。并且除了在 $\gamma = 0$ 处以外基本呈现一条水平线。反观 HP-HC 算法，随着 γ 的变化，HP-HC 算法的 F1 评价指标变化很明显，在 $\gamma = -0.5$ 时的 F1 指标比最低点提高了 150%。图5-7给出了在 Netflix 数据集中对应的结果。在 Netflix 数据集上，HP-MD 算法的推荐准确性随参数 γ 变化的程度依然很小，F1 指标提高幅度小于 2%。而 HP-HC 的变化跟在 MovieLens 数据集上的基本一致，呈现较大幅度的改变，在 $\gamma = -0.77$ 时的 F1 指标比最低点提高了 125%。值得注意的是，在两个数据集上，HP-HC 算法得到最优 F1 指标时的 γ 值都要比 0 小，表明近期比较流行的物品在推荐算法中的重要性被加强了。除了准确率以外，在 MovieLens 数据集中，HP-HC 算法新颖性指标的数值在准确性提高的同时出现了较大幅度的降低，表示推荐列表中包含了更多的小众物品。多样性指标在 $\gamma > 0$ 时几乎保持不变，当 $\gamma < 0$ 时，多样性指标出现了陡降。这说明在 MovieLens 数据集中 HP-HC 是以牺牲多样性指标来提升了准确性。在 Netflix 数据集中，HP-HC 算法的多样性和新颖性显得很平稳，在准确性发生较大变化时并未

表现出明显的变化。HP-HC 算法在不丢失推荐结果多样性水平的情况下，大幅提升了推荐结果的准确性。

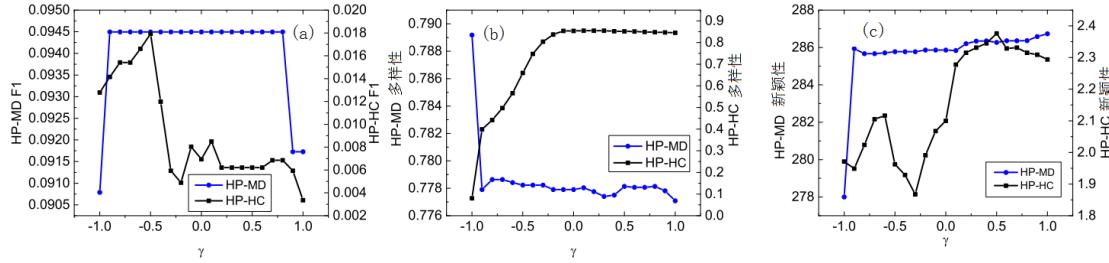


图 5-6 MovieLens 数据集上 HP-MD 和 HP-HC 算法的推荐性能

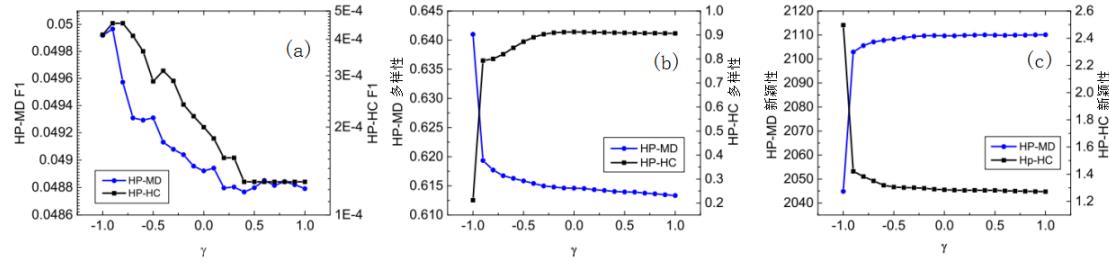


图 5-7 Netflix 数据集上 HP-MD 和 HP-HC 算法的推荐性能

5.4 近期流行度在热传导推荐算法中的研究

通过实验结果可以发现，虽然 HP-HC 算法的 F1 指标在绝对值上依然很低，并不能直接应用在生产环节，但是它的相对提升幅度表明物品的近期流行度对基于热传导的推荐算法的性能提升有很大的帮助作用。相反，将物品近期流行度融合到基于资源分配的算法中并不能带来推荐性能的大幅提升。本节将进一步研究物品近期流行度对热传导类推荐算法的改进作用。

5.4.1 基于近期流行度的热传导算法

目前，研究人员已经提出了很多改进的基于热传导的推荐算法，这些算法可以在保留标准热传导算法高多样性指标的同时显著提升准确率，克服了标准热传导算法过分关注小度物品的问题，出色的解决了准确性和多样性的两难问题。因此，本章将物品近期流行度应用到以下三个基于热传导的推荐算法中，以观察近期流行度是否可以进一步提高这类推荐算法的推荐性能：

(1) HPH 是一个融合了 HC 和 MD 的混合算法，用以解决多样性-准确性困境问题。该模型利用参数 γ 调节 MD 过程和 HC 过程所占的比重，转移矩阵 W 的元

素 $w_{\alpha\beta}$ 定义为:

$$w_{\alpha\beta}^{HPH} = \frac{1}{k_\alpha^{1-\lambda} k_\beta^\lambda} \sum_{i=1}^m \frac{a_{i\alpha} a_{i\beta}}{k_i} \quad (5-7)$$

(2) GHC 是由 Zhou 等人^[161] 提出的一种改进的热传导算法, 该算法添加了一个虚拟 ground 用户节点与所有物品相连, 增加了网络的连通性, 转移矩阵 W 的元素 $w_{\alpha\beta}$ 定义为:

$$w_{\alpha\beta}^{GHC} = \frac{1}{k_\alpha + 1} \sum_{i=1}^{m+1} \frac{a_{i\alpha} a_{i\beta}}{k_i} \quad (5-8)$$

(3) DWC 是由 Ren 等人^[162] 提出的另一种改进的热传导算法, 该方法将每条连边考虑成不同材质的传导介质, 在原有二部图基础上为每条连边增加了方向和权重, 转移矩阵 W 的元素 $w_{\alpha\beta}$ 定义为:

$$w_{\alpha\beta}^{DWC} = \frac{1}{k_\alpha} \sum_{i=1}^m \frac{c_{i\beta} a_{i\alpha} a_{i\beta}}{\sum_{\delta=1}^n c_{i\delta} a_{i\delta}} \quad (5-9)$$

其中, $c_{i\delta} = (k_i k_\delta)^\lambda$, 表示物品到用户的有向边权重。

通过公式 (5-4) 将物品近期流行度融合到上述三种热传导算法中, 分别记为 HP-HPH、HP-GHC 和 HP-DWC。融合后的各算法中转移矩阵元素 $w_{\alpha\beta}(t^*)$ 分别定义为:

$$w_{\alpha\beta}^{HP-HPH}(t^*) = \frac{\sum_{i=1}^m a_{\alpha\alpha}(t^*) a_{i\beta}(t^*) / k_i(t^*)}{[k_\alpha(t^*) + \gamma \Delta k_\alpha(t^*, T_P) + \epsilon]^{1-\lambda} [k_\beta(t^*) + \gamma \Delta k_\beta(t^*, T_P) + \epsilon]^\lambda} \quad (5-10)$$

$$w_{\alpha\beta}^{HP-GHC} = \frac{1}{k_\alpha(t^*) + \gamma \Delta k_\alpha(t^*, T_P) + \epsilon + 1} \sum_{i=1}^{m+1} \frac{a_{i\alpha}(t^*) a_{i\beta}(t^*)}{k_i(t^*)} \quad (5-11)$$

$$w_{\alpha\beta}^{HP-DWC} = \frac{1}{k_\alpha(t^*) + \gamma \Delta k_\alpha(t^*, T_P) + \epsilon} \sum_{i=1}^m \frac{c_{i\delta}'(t^*) a_{i\alpha}(t^*) a_{i\beta}(t^*)}{\sum_{\delta=1}^n c_{i\delta}'(t^*) a_{i\delta}(t^*)} \quad (5-12)$$

其中, $c_{i\delta}'(t^*) = (k_i(t^*) (k_\delta(t^*) + \gamma \Delta k_\delta(t^*, T_P) + \epsilon))^\lambda$ 。

为了将重点放到分析近期流行度对推荐性能的影响上, HP-HPH 和 HP-DWC 算法中的参数 λ 设置为 HPH 和 DWC 算法得到的最优 F1 时对应的参数值, 即忽略参数 λ 对 HP-HPH 和 HP-DWC 的影响。

5.4.2 基于近期流行度的热传导算法性能分析

本章首先研究控制近期流行度所占比例的参数 γ 对基于近期流行度的热传导算法的作用。图5-8和图5-9分别显示了在 MovieLens 和 Netflix 两个数据集上,

HP-HPH、HP-GWC 和 HP-DWC 的 F1 指标、多样性和新颖性随参数 γ 的变化情况。当 $\gamma = 0$ 时，可以得到标准的 HPH、GWC 和 DWC 算法的推荐结果。在 MovieLens 中，HP-HPH 和 HP-DWC 算法的 λ 分别等于 0.4 和 -1.9；在 Netflix，HP-HPH 和 HP-DWC 算法的 λ 分别等于 0.6 和 -1.0。

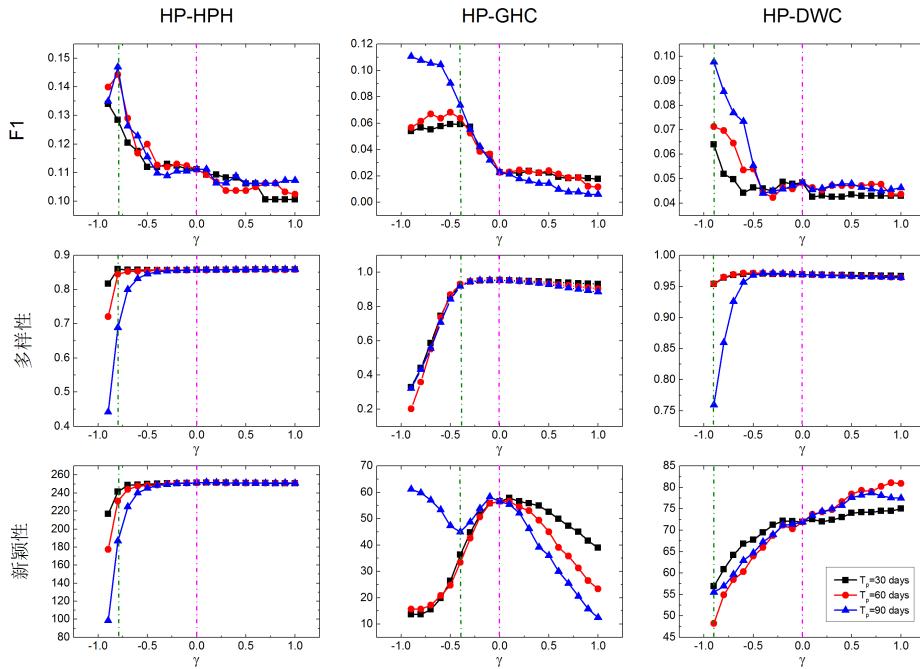


图 5-8 MovieLens 数据集上 HP-HPH、HP-GHC 和 HP-DWC 算法的推荐性能

如图5-8和图5-9所示，对于 HP-HPH、HP-GWC 和 HP-DWC 三个算法，在 $\gamma < 0$ 时的 F1 指标都要比 $\gamma = 0$ 时高很多，表明融合了近期流行度的热传导算法的推荐准确性都要优于原始算法。这也进一步支持了本章的假设，即通过结合物品近期流行度可以增加近期热门物品的推荐概率，改进基于热传导的推荐算法的推荐效果。但当参数 γ 过小时，推荐结果的多样性要明显低于 $\gamma = 0$ 时。为了解决多样性-准确率困境，本章通过遍历参数 γ ，以经验启发式的方法将 HP-HPH、HP-GHC 和 HP-DWC 在 MovieLens 数据集上的参数 γ 选定为 -0.8、-0.4 和 -0.9，类似的，将在 Netflix 数据集上的参数 γ 选定为 -0.6、-0.5 和 -0.4。表5-2总结了在 $T_P = 60$ 时两个数据集上的推荐结果。从表中可以发现，相较于 HPH、GHC 和 DWC 算法，HP-HPH、HP-GHC 和 HP-DWC 在 MovieLens 数据集上的准确性可以分别提高 29.88%、178.95% 和 47.31%，而在 Netflix 数据集上，HP-HPH、HP-GHC 和 HP-DWC 的准确性可以分别提高 26.25%、445.76% 和 54.88%。与此同时，通过观察和比较

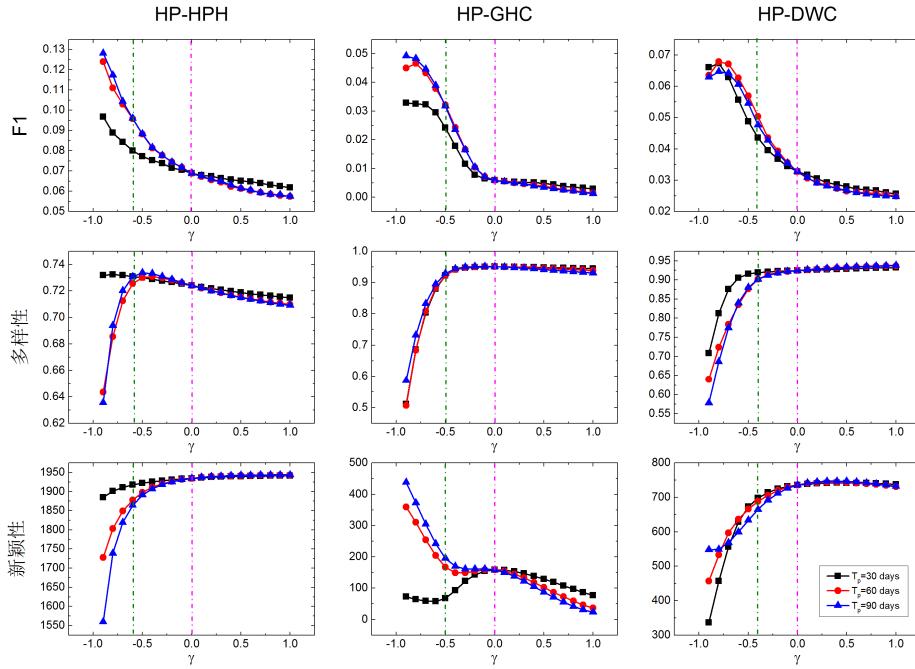


图 5-9 Netflix 数据集上 HP-HPH、HP-GHC 和 HP-DWC 算法的推荐性能

图5-8和图5-9中融合算法的最优值（左侧绿色虚线处）和 $\gamma = 0$ 时的原始算法结果（右侧紫色虚线处）可以看到，基于近期流行度算法的多样性指标只比原始算法降低了很小的程度。融合算法的新颖性指标要远低于原始算法的结果，新颖性数值的降低代表融合了物品近期流行度的算法可以帮助推荐系统发现更加小众的物品，一定程度上解决了长尾问题。通过准确性和多样性两个角度的分析，融合了近期流行度的热传导算法可以比原始算法提供更准确和更加个性化的推荐结果。与第三章提出的 TaCF 和第四章提出的 TaDb 算法相比，融合近期流行度的 HP-HPH 算法在准确性方面得到了显著提升，而 HP-GHC 和 HP-DWC 算法在个性化方面则大幅提高了推荐结果的多样性和新颖性。

其次，在图5-5中发现时间窗口长度 T_P 和物品间的 Pearson 相关性有着某种关系。为了更好的理解时间窗口长度 T_P 的选取对基于近期流行度的推荐算法结果的影响，本章对不同时间窗口长度下推荐算法的推荐能力进行了分析。为了进行对比，时间窗口长度 T_P 依次选取了三个数值，分别是 30 天、60 天和 90 天，实验评价指标选取 F1 指标、多样性和新颖性。图5-8和图5-9中不同颜色的曲线代表了 3 种时间窗口长度下的推荐结果，黑色代表 30 天、红色代表 60 天、蓝色代表 90 天。在 $\gamma < 0$ 的一段很大范围内（两条虚线间的区域），无论 T_P 取

30、60 还是 90 天, MovieLens 和 Netflix 数据集中的 F1 指标都要高于当 $\gamma = 0$ 时的数值, 此时推荐结果的多样性相较于 $\gamma = 0$ 时几乎没有什么明显变化。此外, 在两个数据集中, 当 $T_p = 90$ 天时得到的 F1 指标并不差于 $T_p = 30$ 和 60 天时的结果, 而 $T_p = 90$ 时得到的多样性是最低的。图5-5显示 Netflix 数据集上存在 $PCC_{T_p=30} > PCC_{T_p=60} > PCC_{T_p=90}$ 的单调关系, 通过比较不同窗口长度下的评价指标可以得出推荐结果与 PCC 之间不存在简单的线性相关性。如何选取一个合适的时间窗口长度仍然是需要进一步研究的问题。

表 5-2 MovieLens 和 Netflix 数据集上不同算法的性能比较

| 算法 | MovieLens | | | Netflix | | |
|--------|---------------|---------------|---------------|---------------|---------------|----------------|
| | F1 | 多样性 | 新颖性 | F1 | 多样性 | 新颖性 |
| HPH | 0.1111 | 0.8568 | 251.50 | 0.0499 | 0.7241 | 1934.85 |
| HP-HPH | 0.1443 | 0.8448 | 231.05 | 0.0630 | 0.7255 | 1878.36 |
| GHC | 0.0228 | 0.9536 | 56.60 | 0.0059 | 0.9506 | 158.33 |
| HP-GHC | 0.0636 | 0.9304 | 33.37 | 0.0322 | 0.9223 | 167.03 |
| DWC | 0.0484 | 0.9688 | 71.92 | 0.0328 | 0.9244 | 735.94 |
| HP-DWC | 0.0713 | 0.9540 | 48.21 | 0.0508 | 0.9015 | 688.29 |
| TaDb | 0.0740 | 0.7072 | 629.54 | 0.0495 | 0.5299 | 965.20 |
| TaCF | 0.0794 | 0.8634 | 230.57 | 0.0456 | 0.8264 | 2092.41 |

不同的用户对最近流行物品可能表现出明显不同的偏好, 最近流行的物品对大度用户可能并没有有很强的吸引力。本章将用户根据不同的度分成不同的组别, 以研究最近流行度对不同用户的推荐作用。为了分析物品最近流行度在不同组别中的影响力, 分别用 HP-HPH、HP-GHC 和 HP-DWC 算法为每名用户产生推荐列表并计算 F1 指标, 然后将组别内不同用户的平均 F1 指标作为该组别的推荐结果。图5-10显示了在 MovieLens 数据集上不同组别中的 F1 指标结果, 从中可以发现, 相对于原始算法, 融合了近期流行度的算法可以有效的提高小度和大度用户的推荐准确性, 而对中度用户的作用并不明显。图5-11显示了在 Netflix 数据集上对应的结果。不同于 MovieLens 数据集上的结果, Netflix 数据集中的结果表明融合了近期流行度的推荐算法对所有组别的用户都可以显著提高推荐准确性。此外, 根据统计结果显示, Netflix 数据集中大多数用户的度都落在 1 到 100 的范围内, 这意味着基于近期流行度的热传导算法能够帮助绝大部分用户找到更适合自己的物品。

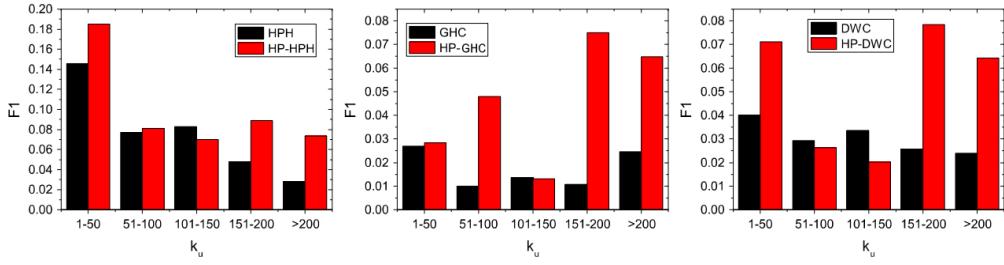


图 5-10 MovieLens 数据集不同用户组的推荐准确率

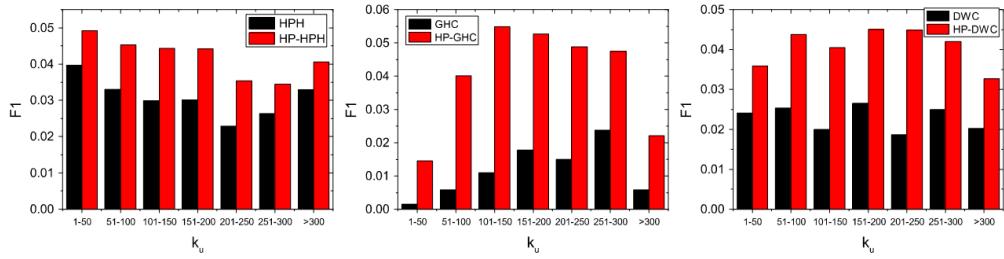


图 5-11 Netflix 数据集不同用户组的推荐准确率

5.5 基于近期流行度的后处理算法

虽然将近期流行度融合到基于资源分配的算法中没有得到预期的效果，但是通过基于近期流行度的热传导算法可以发现增加近期流行物品的权重确实可以进一步提高推荐准确性。为了进一步分析物品近期流行度的作用，本章提出了一种基于近期流行度的后处理算法，将基于资源分配的推荐算法得出的物品评分进行了重新处理。

5.5.1 后处理算法在资源分配算法中的应用

针对目标用户 u_i ，基于资源分配的推荐算法最终会为每个物品 o_α 分配一定数量的资源，表示为用户选择该物品的概率值或者是用户对物品的预测评分。为了突出近期流行物品的作用，本章使用物品近期流行度与总流行度的比值作为衡量物品未来流行趋势的概率：

$$pop_\alpha = \frac{\Delta k_\alpha(t^*, T_p)}{k_\alpha(t^*)} + \epsilon \quad (5-13)$$

其中， ϵ 是为了避免物品近期没有任何人购买，导致 $pop_\alpha = 0$ 失去了被推荐的可能，本章将 ϵ 设置为 10^{-2} 。 pop_α 的值表示物品近期的流行度占整个生命周期中的比例，值越大表示物品在近期越可能成为热门物品。后处理技术是基于上下文信息的推荐算法中最重要的一种融合上下文信息和推荐算法的手段，本章将资源分

配算法得到的物品最终资源 f_α' 乘以 pop_α 对物品的评分进行重新处理，物品 o_α 的最终排序分等于：

$$r_\alpha = f_\alpha' \times pop_\alpha^\lambda = f_\alpha' \left(\frac{\Delta k_\alpha(t^*, T_P)}{k_\alpha(t^*)} + \epsilon \right)^\lambda \quad (5-14)$$

其中， λ 是一个调节参数，在不同的系统中，物品近期流行度所占的比重可能是不同的，通过参数 λ 可以调节近期流行度在不同数据集中的作用，当 $\lambda = 0$ 时可以得到未经后处理的原始算法结果。最后，用户未购买过的物品按照排序分的大小降序排列，将排名靠前的 L 个物品推荐给用户。

5.5.2 后处理算法的性能分析

在不同的系统中，用户对近期流行的物品的敏感程度是不一样的。例如，在新闻系统中，用户更倾向选择近期热门的内容；而在旅游行业，过分关注热门商品会导致人群出现在同一个地点，用户无法获得优质的体验。为此，本章首先分析参数 λ 对推荐性能的影响。图5-12给出了在 MovieLens 和 Netflix 两个数据集中推荐结果的各项指标随参数 λ 的变化情况，实验过程中 T_P 设置为 10 天。

首先从准确性方面来分析后处理算法推荐性能与参数 λ 的关系。为了全面了解准确性的具体情况，我们使用了准确率、召回率和 F1 指标三个评价标准。如图5-12所示，在 MovieLens 数据集上，准确率、召回率和 F1 指标都是先随着参数 λ 的增加而增大，其中准确率和 F1 指标在经过极值点后出现迅速下降，而召回率经过极值点后仍保持相对较高的数值。以 F1 指标为主要评价指标，参数 λ 的最优值出现在 0.8 附近。此时，召回率仍可以继续提高，但是准确率开始快速下降。相比于 $\lambda = 0$ 的标准资源分配算法，后处理算法使得推荐结果在准确率、召回率和 F1 指标上都有了显著的提高，其中准确率提高了 10.3%，召回率的提升比例超过了 100%，F1 指标提高了 54.5%。在 Netflix 数据集上，准确率、召回率和 F1 指标都呈现出随着参数 λ 先上升后下降的趋势，最优 F1 指标对应的参数 $\lambda = 1.3$ 。在最优 F1 指标处，准确率、召回率和 F1 指标相比于 $\lambda = 0$ 处分别提高了 87.1%、200%、117%。从结果可以发现，经过后处理的推荐准确性要明显优于原始算法，不仅能够提高推荐准确率还能发现用户更多的潜在兴趣。对比 MovieLens 和 Netflix 数据集上的结果可以发现，MovieLens 上最优 F1 对应的 $\lambda < 1$ ，而 Netflix 上的 $\lambda > 1$ ，这表明 MovieLens 上的用户更倾向选择热门物品。

接着，从多样性、覆盖率和新颖性三个方面分析后处理算法的推荐性能。从图5-12可以发现，不管是 MovieLens 还是 Netflix，都是以牺牲推荐结果的多样性来提高推荐准确性。相对于 $\lambda = 0$ 的原始结果，经过后处理的推荐结果牺牲了比较大的多样性指标。虽然损失了用户之间的推荐列表差异性，但是被推荐物品的

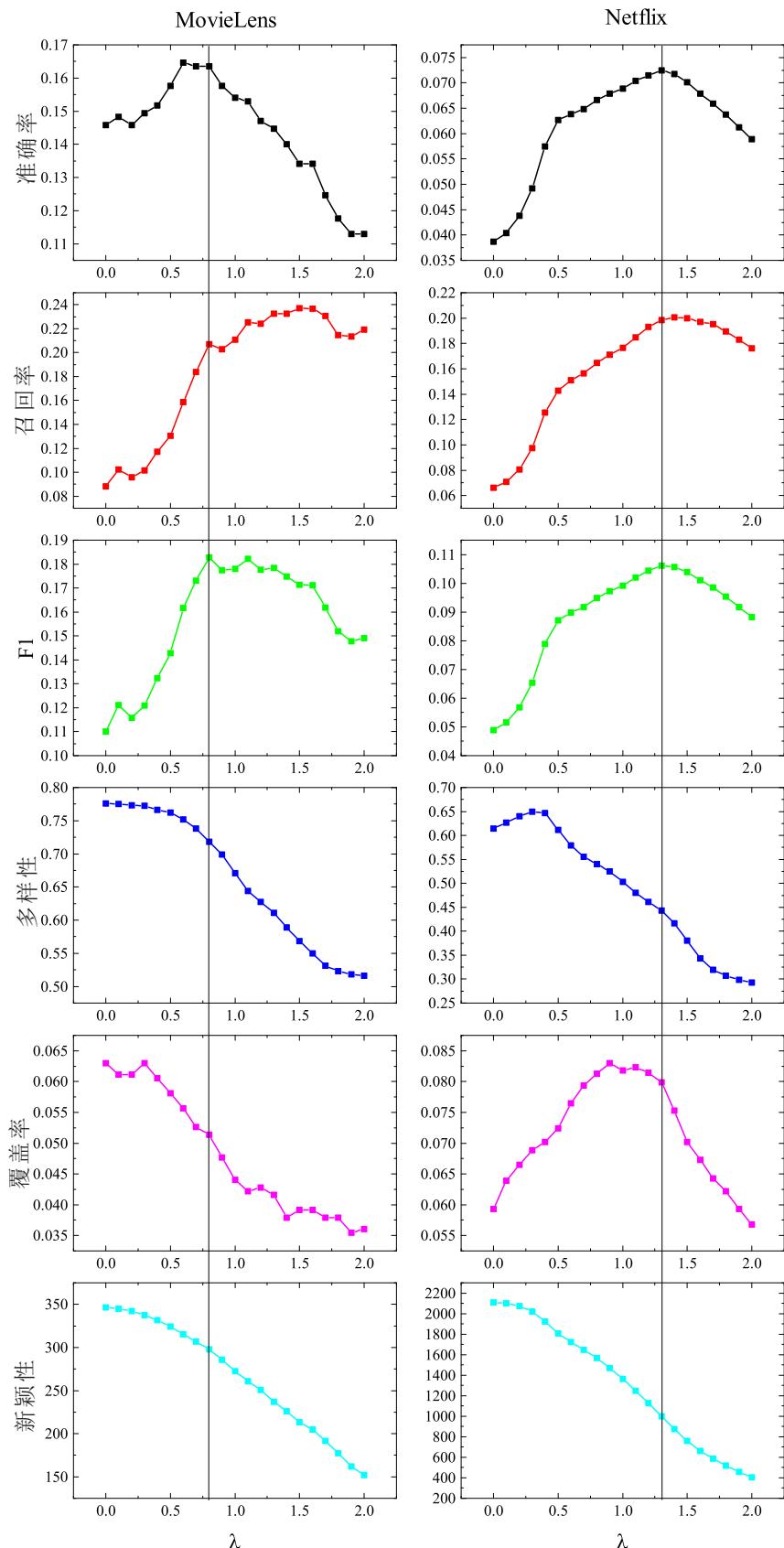


图 5-12 MovieLens 和 Netflix 数据集上后过滤处理的推荐性能

平均流行度明显的降低了，算法推荐了更多的小众物品给用户。覆盖率方面，在 Netflix 数据集上，最优 F1 指标对应的覆盖率要比原始结果高出 30% 以上，而在 MovieLens 数据集上，覆盖率损失了大约 16.6%。

综合准确性、多样性、覆盖率和新颖性评价指标可以发现，经过后处理的推荐算法可以为用户推荐更多的小众商品，而这些小众商品正是用户喜欢的，这能够帮助系统发掘更多长尾商品，具有很强的商业应用价值。

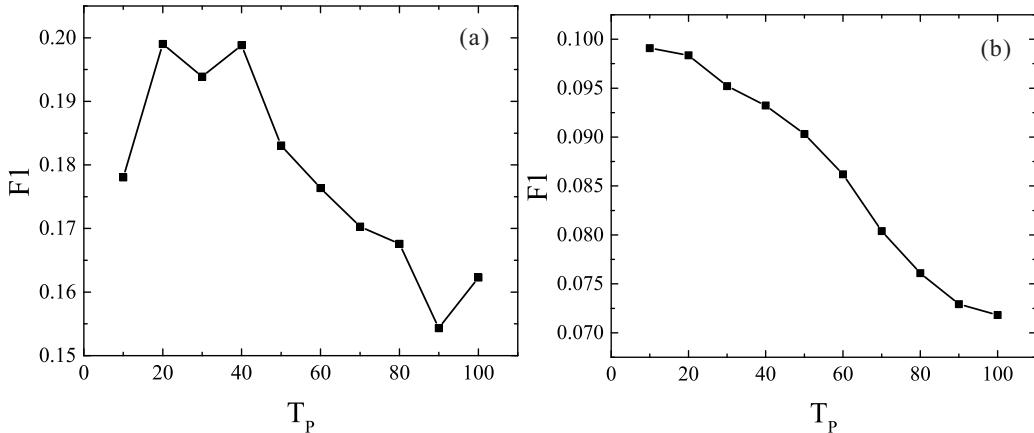


图 5-13 不同时间窗口长度下的 F1 指标。(a) MovieLens; (b) Netflix

基于近期流行度的后处理算法除了参数 λ 以外，还有一个重要的参数，即时间窗口长度 T_P 。 T_P 越大包含的历史数据越多，可能会淹没近期真正流行的物品； T_P 太小可能会使得包含的信息太少无法准确衡量物品的近期流行性。因此，接下来分析参数 T_P 对后处理算法的推荐性能影响。图5-13给出了在 MovieLens 和 Netflix 两个数据集上 F1 指标随参数 T_P 的变化情况。从图中可以看出，在两个数据集上，F1 指标的总体趋势是随着 T_P 的增加而逐渐降低，表明物品的近期流行度要比物品的长期流行度更为重要，考虑较长时间周期内的物品流行度会使得系统推荐很多过时的物品。但是 T_P 也不能过小，例如在 MovieLens 数据集上， $T_p = 10$ 的结果要明显低于 $T_p = 20$ 的结果，这表明过小的时间窗口长度会损失一部分物品的近期流行度，导致对物品的流行度趋势刻画的不够准确。

5.6 本章小结

本章首先从物品的角度分析了系统表现出的一些时间特性。然后，基于物品的近期流行度，提出了融合近期流行度的资源分配推荐算法和融合近期流行度的热传导推荐算法。实验结果表明，相较于资源分配算法，融合了近期流行度的热传导推荐算法更能提高推荐算法的性能。为了进一步研究近期流行度的作用，本章将近期流行度融合到三个基于热传导的推荐算法中。通过在 MovieLens 和 Netflix

数据集上的实验结果表明，融合了近期流行度的热传导推荐算法可以进一步提高原有推荐模型的准确性和新颖性，同时保留原有模型高多样性的优点。本章还分析了近期流行度对不同用户的影响，发现基于近期流行度的推荐算法可以帮助系统中大部分用户找到更合适的物品，尤其是对小度用户，这一定程度上解决了系统冷启动问题。最后，本章提出了一种基于近期流行度的后处理算法，通过将资源分配算法得出的物品评分根据近期流行度重新加权处理后，推荐性能得到了显著的提高，算法能够发掘更多用户喜欢的长尾商品，具有较高的商业应用价值。

第六章 基于相对时间的推荐算法研究

6.1 问题的提出

目前，时间相关的推荐算法研究主要关注的是用户行为的绝对时间，利用绝对时间可以分析行为的时间间隔、用户和物品的生命周期、距离当前时刻的时间衰减作用等。但是在早期的互联网系统设计过程中，由于时间信息对分析用户行为的重要性还没有得到足够认识，很多记录的时间信息并没有得到很好的保存，在有些推荐场景中无法得到用户行为的精确时间信息。在数据库系统中，主键是记录的唯一识别 ID，很多数据库的主键类型采用了自增类型，其值会根据记录的添加自动递增。因此，即使没有记录的绝对时间，我们还是可以得到记录产生的相对时间，即记录的先后顺序。根据这些信息，通常可以得到物品序列和用户序列两种相对时间关系。物品序列指同一用户选择不同物品的先后顺序；用户序列指同一物品先后被哪些用户选择了。

在推荐系统中，用户对物品的评分顺序可以很好的反映用户偏好。如图6-1所示，假设目标用户分别为物品 1、物品 2 和物品 3 评了 4 分、3 分和 2 分。通过将评分进行排序，可以得到用户的偏好顺序是：物品 1> 物品 2> 物品 3。现在有两个推荐算法，算法 1 为目标用户针对物品 1、物品 2 和物品 3 的预测评分是 3 分、4 分和 3 分；算法 2 的预测结果是 3 分、2 分和 1 分。在这种情况下，两个算法的预测误差平方和是相同的，但是预测结果的偏好顺序却是不同的。可以很明显的发现算法 2 的预测结果表现出的偏好顺序与目标用户是一致的。使用算法 2 可以为用户提供更加满意的推荐列表。用户的评分顺序除了能够表征个体的偏好顺序，还能反映用户与用户之间的相似性。把上例中的两种算法看成是普通用户，虽然用户 2 都在为物品打负分，但是他与目标用户的偏好要更为一致些。

| | 物品1 | 物品2 | 物品3 |
|---------|-----|-----|-----|
| 目标用户 | 4 | 3 | 2 |
| 算法1/用户1 | 3 | 4 | 3 |
| 算法2/用户2 | 3 | 2 | 1 |

图 6-1 用户评分偏好顺序样例

利用相对时间得到的物品序列和用户序列能否同评分顺序一样表现出用户的偏好信息，这是本章研究的主要内容之一。在隐性反馈系统中，基于图的推荐算法被认为是比传统协同过滤算法更好的算法，能够提供更好的推荐准确性，并且计

算复杂度更低。能否将用户行为的相对时间信息结合到基于图的推荐算法中，提高推荐系统的各项指标，这是本章研究的另一个内容。

6.2 基于用户序列的资源分配算法

通过选择同一物品的用户序列，可以得到购买该物品的每两个用户之间的先后顺序。依次遍历所有物品后，可以很容易的得到每两个用户在购买相同商品中的相对顺序。这种顺序关系可以用来衡量两个用户之间的相关性。本章假设相关性越强的用户购买的物品越能表示目标用户未来的兴趣。将用户之间的相关性融合到基于资源分配算法中提出了一种基于用户序列的资源分配算法（User-Sequence based Diffusion algorithm，记为 USD），该算包括两阶段：第一个阶段是用户间的相关性计算，第二个阶段是偏好预测。

6.2.1 相关性计算

用户 u_j 对用户 u_i 的影响最直观的衡量方式是看用户 u_i 跟随用户 u_j 共同购买过的物品数量，以及用户 u_i 跟随用户 u_j 购买物品时的延迟时间 Δt 。用户 u_i 跟随用户 u_j 购买的物品数量越多并且延迟时间越短，代表用户 u_i 越信任用户 u_j ，受用户 u_j 的影响也就越大。类似的，用户 u_i 对用户 u_j 的影响也可以用类似的概念来衡量。在只有相对时间的情况下，本章直接使用用户 u_i 跟随用户 u_j 购买的物品数量占用户 u_j 购买的物品数量的比例作为用户 u_j 对用户 u_i 的影响，定义为：

$$Inf_{j,i} = \frac{|O_{j,i}|}{k_j} \quad (6-1)$$

其中， $O_{j,i}$ 表示用户 u_i 跟随用户 u_j 购买的物品集合， k_j 表示用户 u_j 的度，即用户 u_j 购买的物品数量。用户间的相关性应该是彼此之间的相互影响，而不是单方面的。因此，本章把用户 u_j 和用户 u_i 对彼此的影响进行线性融合，将其定义为用户间的相关性，公式表示为：

$$\begin{aligned} C_{ij} &= \alpha Inf_{i,j} + (1 - \alpha) Inf_{j,i} \\ &= \alpha \frac{|O_{i,j}|}{k_i} + (1 - \alpha) \frac{|O_{j,i}|}{k_j}, \text{ 如果 } \frac{|O_{i,j}|}{k_i} \geq \frac{|O_{j,i}|}{k_j} \end{aligned} \quad (6-2)$$

其中，参数 α 表示影响力较大的用户所占的比重。如图6-2所示，在三个物品的用户序列中，用户 u_1 购买了 3 个物品，用户 u_3 购买了 2 个物品。在购买物品 1 时，用户 u_1 在用户 u_3 之前购买，而用户 u_3 比用户 u_1 先购买了物品 3。用户 u_1 对用户 u_3 的影响可以简单的记为 $1/3$ ，而用户 u_3 对用户 u_1 的影响等于 $1/2$ 。通过线性加

权后他们的相关性等于 $\alpha/6 + 1/3$ 。

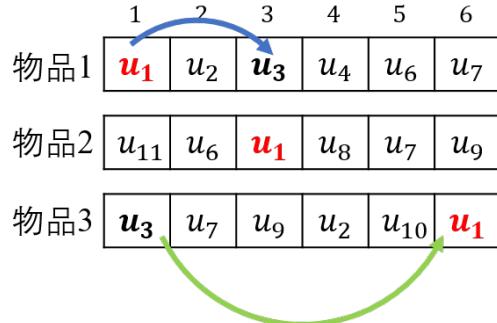


图 6-2 用户序列样例

6.2.2 偏好预测

标准的基于资源分配算法假设用户的偏好可以以资源或能量的形式通过二部图网络进行扩散，扩散的过程可以使用转移矩阵 W 表示，其中元素 $w_{\alpha\beta}$ 表示物品 o_α 从物品 o_β 处获得的能量，定义为：

$$w_{\alpha\beta} = \frac{1}{k_\beta} \sum_{l=1}^m \frac{a_{l\alpha}a_{l\beta}}{k_l} \quad (6-3)$$

从公式 (6-3) 可以发现，在标准的基于资源分配的推荐算法中，物品 o_α 从物品 o_β 处获得的资源仅依赖于物品 o_β 的度和那些同时选择过物品 o_α 和 o_β 的用户的度。该方法并未考虑用户之间的相关性，仅仅利用了局部信息。即使两个用户的度可能相同，并且选择过很多相同的物品，但是他们可能对这些物品的购买顺序是截然不同的，对这些物品的偏好就可能存在巨大差别。

在向目标用户 u_i 产生推荐结果时，将用户-物品二部图中每条连边 e_{ja} 根据该连边关联的用户 u_j 和目标用户 u_i 的相关性作为连边权重，当偏好在网络中扩散时，转移矩阵 W 中的元素定义 $w_{\alpha\beta}$ 为：

$$w_{\alpha\beta}^i = \frac{1}{k_\beta} \sum_{l=1}^m \frac{(a_{l\alpha}c_{il})(a_{l\beta}c_{il})}{k_l} \quad (6-4)$$

6.3 基于物品序列的资源分配算法

从用户的角度出发，可以根据数据库中主键的先后顺序，得到每个用户购买过的物品的相对顺序。本章利用用户间共同购买的物品序列关系来衡量用户的相似度，提出了一个基于物品序列的资源分配算法（Item-Sequence based Diffusion

algorithm, 记作 ISD), 该算法包括两个阶段: 第一阶段是相似度计算, 第二阶段是偏好预测。该算法首先计算每两个用户之间的相似度, 然后为目标用户预测将来的偏好并推荐未购买过的商品, 在进行预测时用户间的相似度被作为用户-物品二部图网络中的连边权重, 以此增加相似用户之间的影响。

6.3.1 相似度计算

ISD 算法首先使用 Plackett-Luce 模型 (一个广泛使用的排列概率模型) 将每个用户表征为一个概率分布向量, 向量中每一个元素表示由该用户购买过的物品组成的一种排列的概率值。算法假设这些物品组成的任何排列都是可行的, 且不同的排列有不同的概率, 越接近“正确的”排列 (即用户真实偏好的顺序) 的概率越大。通常这种排列顺序可以使用用户的评分信息进行推断, 但是本章研究的背景是在隐性反馈数据下。因此, 本章用相对时间来辅助确认用户的偏好信息。简单起见, 可以认为用户最近购买的物品更为重要, 更能代表用户当前的兴趣, 为此给予这些物品更高的贡献分。针对用户 u_i , 物品 o_α 在该用户购买的物品组成的排列中的贡献分 r_i^α 定义为:

$$r_i^\alpha = e^{-\lambda(k_i - L_{i\alpha})} \quad (6-5)$$

其中, k_i 为用户 u_i 的度, $L_{i\alpha}$ 为用户 u_i 购买的物品 o_α 在用户 u_i 的物品序列中的位置, 参数 λ 是一个时间衰减因子。通过公式 (6-5) 可以看出, 越排在物品序列前端的物品, 其贡献分越小, 排在最后一个物品的贡献分等于 1。

集合 O 上的一个排列可以理解为从 O 到它自身的双射关系。排列可以用一个有序列表 $\pi = \langle \pi_1, \pi_2, \dots, \pi_n \rangle$ 表示, 其中 $\pi_\alpha \in O$ 表示排在第 α 位的物品。一个物品不会同时出现在列表的两个位置中, 即当 $\alpha \neq \beta$ 时, $\pi_\alpha \neq \pi_\beta$ 。集合 O 中物品的所有排列构成的集合标记为 Ω^O 。很明显, 对于包含 n 个物品的集合, 会有 $n!$ 种不同的排列。因此, 要计算所有排列的概率值将会是十分耗时的工作。由于实际系统中用户和物品的交互行为十分频繁, 使用全部 n 个物品会使得计算出的结果对用户没有任何实际价值, 得到计算结果时用户的兴趣偏好可能已经发生了改变。为了解决该问题, 本章使用了 Top-k 概率模型^[163]。该模型只关注排在列表前 k 个位置的物品组成的各种可能性。给定一个物品集合 O , $\mathcal{G}_k(o_1, o_2, \dots, o_k)$ 表示所有以 o_1, o_2, \dots, o_k 的物品顺序为开头的排列构成的集合, 可用数学公式表示为:

$$\mathcal{G}_k(o_1, o_2, \dots, o_k) = \{\pi | \pi \in \Omega^O, \pi_\beta = o_\beta, \beta = 1, 2, \dots, k\} \quad (6-6)$$

因此, Top-k 排列集合的概率 $P(\mathcal{G}_k(o_1, o_2, \dots, o_k))$ 等于物品 o_1, o_2, \dots, o_k 依次

排在前 k 个位置的概率，公式定义为：

$$P(\mathcal{G}_k(o_1, o_2, \dots, o_k)) = \prod_{\beta=1}^k \frac{\Phi(r_{\pi_\beta})}{\sum_{l=\beta}^n r_{\pi_l}}, \forall \beta = 1, 2, \dots, k : \pi_\beta = o_\beta \quad (6-7)$$

其中， $\Phi(r) = e^r$ ，是一个严格为正的递增函数，而 r_{π_l} 是使用公式 (6-5) 计算的排在第 l 个位置的物品的贡献分。贡献分越高的物品表示用户越喜欢该物品，排在列表前端的概率也就越大。

针对每一对用户 u_i 和 u_j ，让 O_{ij} 表示他们共同购买过的物品集合。符号 $\mathcal{G}_k^{O_{ij}}$ 表示集合 O_{ij} 上的 Top-k 排列集合的集合，例如用户 u_i 和 u_j 有 5 个共同购买的物品，则 $\mathcal{G}_k^{O_{ij}}$ 是一个包含 A_5^3 个 Top-3 排列集合的集合。 P_i 和 P_j 分别是以 $\mathcal{G}_k^{O_{ij}}$ 中所有排列为元素表示的用户 u_i 和 u_j 的概率分布向量。用户 u_i 和 u_j 之间的相似度可以通过 P_i 和 P_j 两个概率分布进行估算。本章使用基于 Jensen-Shannon 散度指标（简称 JS）的方法计算用户间相似度，在概率理论和信息化理论中，该指标是衡量两个概率分布间相似度的常用方法。给定两个用户 u_i 和 u_j ，对应的 P_i 和 P_j 的 JS 散度定义为：

$$D_{JS}(P_i||P_j) = \frac{1}{2}D_{KL}(P_i||M) + \frac{1}{2}D_{KL}(P_j||M) \quad (6-8)$$

其中， $M = (P_j + P_i)/2$ ， $D_{KL}(P_i||M)$ 表示分布 P_i 到分布 M 的 Kullback-Leibler (KL) 散度。KL 散度的定义为：

$$D_{KL}(P_i||M) = \sum_{g \in \mathcal{G}_k^{O_{ij}}} P_i(g) \log_2 \left(\frac{P_i(g)}{M(g)} \right) \quad (6-9)$$

其中， $P_i(g)$ 表示 Top-k 排列集合 $g \in \mathcal{G}_k^{O_{ij}}$ 对于用户 u_i 的概率。 $D_{KL}(P_j||M)$ 可以使用相似的方法计算得到。为了衡量用户间的相似度，本章使用了基于 JS 散度的对称相似度衡量指标，定义为：

$$s(i,j) = 1 - \sqrt{D_{JS}(P_i||P_j)} \quad (6-10)$$

该相似度存在一个问题，即如果两个用户只购买了很少的共同物品，该相似度值往往会偏大。为了克服该问题，本章使用文献 [164] 提出的方法，将计算得到的相似度乘以 $\min\{|O_{ij}|/c, 1\}$ ，其中 c 是一个阈值。当共同购买过的物品数量小于阈值时，相似度数值会按比例缩小。

6.3.2 偏好预测

定义 $S \in R^{m \times m}$ 为经过相似度计算得到的用户相似度矩阵, 矩阵元素 $s_{ij} = s(i, j)$ 。ISD 算法是一种基于用户-物品二部图的资源分配算法, 该方法根据用户相似度矩阵为二部图上的连边赋予不同的权重。例如, 对于目标用户 u_i , 用户节点 u_j 与物品节点 o_α 之间的权重定义为 $a_{j\alpha} \cdot s_{ij}$ 。类似于基于用户序列的资源分配算法, 对于目标用户 u_i , 转移矩阵 W 中的元素 $w_{\alpha\beta}$ 等于:

$$w_{\alpha\beta}^i = \frac{1}{k_\beta} \sum_{l=1}^m \frac{(a_{l\alpha}s_{il})(a_{l\beta}s_{il})}{k_l} \quad (6-11)$$

其中, k_β 是物品 o_β 的度, k_l 是用户 u_l 的度。根据资源分配的过程, 最终物品 o_α 获得的资源等于:

$$p_\alpha^i = \sum_{\beta=1}^n w_{\alpha\beta}^i a_{i\beta} \quad (6-12)$$

用户 u_i 所有未选择过的物品按照最终获得的资源降序排列, 数值较高的物品被推荐给目标用户。

6.4 算法性能评测

6.4.1 实验数据集和对比算法

本章使用了两个电影评分数据集 MovieLens 和 MovieTweetings。MovieLens 采用了最小的 MovieLens 100k 数据集。MovieTweetings 选用了数据量为 100k 的 snapshot, 首先移除评分数量少于 10 个的电影, 然后剔除了评分个数少于 10 个的用户, 处理后的 MovieTweetings 数据集与 MovieLens 数据集的稀疏度在一个数量级上。在两个数据集中, 都忽略了电影评分信息, 只考虑行为发生的先后顺序作为附加信息。具体的统计信息见表6-1所示。

表 6-1 数据集基本统计信息

| 数据集 | 用户数 | 物品数 | 记录数 | 稀疏度 |
|----------------|------|------|-------|----------------------|
| MovieLens | 943 | 1682 | 10000 | 6.3×10^{-2} |
| MovieTweetings | 2581 | 1556 | 52033 | 1.3×10^{-2} |

本章将最后一个自然月中的数据作为测试集, 而发生在这之前的剩余数据作为训练集, 然后从测试集中剔除了未在训练集中包含的用户, 以此模拟真实系统环境下严格利用历史数据预测未来的场景。在 MovieLens 数据集上阈值 c 设置为 20, 在 MovieTweetings 数据集上设置为 10, 两个数据集中参数 λ 都设置成 0.01。

为了分析本章提出的 USD 和 ISD 算法的推荐性能，使用了三个基准算法作为对比，分别是标准的基于资源分配的推荐算法 NBI^[77]、基于资源分配和热传导的混合算法 HPH^[80] 和有偏的热传导算法 BHC^[81]。HPH 是由 Zhou 等人提出的结合资源分配和热传导的混合算法，通过一个参数来调节两个过程所占的比重。实验结果表明该算法可以显著提高推荐准确率和推荐多样性。BHC 算法是由 Liu 等人基于热传导算法提出的一种可以调节算法中大度和小度物品所占比重的改进算法。该算法同样可以得到较高准确率和多样性的推荐结果。

6.4.2 基于用户序列的资源分配算法性能分析

首先，本章将研究在基于用户序列的资源分配推荐算法 USD 中参数 λ 对推荐性能的影响，参数 λ 表示任意两个用户间影响力较大者的重要性。图6-3给出了在 MovieLens 和 MovieTweetings 数据集上推荐结果的准确率、多样性、覆盖率和新颖性随参数 λ 的变化情况。从图中可以发现，MovieLens 数据集上的结果与 MovieTweetings 数据集上的结果较为相似。在准确率方面，都是先随参数 λ 的增加而上升，经过极值点后表现出缓慢的下降趋势。USD 算法的多样性、覆盖率和新颖性指标随参数 λ 的增加都呈现逐渐降低的趋势。在两个数据集中，最优准确率对应的参数 λ 都小于 0.5，表示在推荐过程中影响力较小的用户占据比较大的比重。影响力较小的用户占据主导因素一般会使系统中长尾部分的物品得到更多的推荐可能，从而使系统的个性化水平得到提升。因此，本章接下来将分析 USD 算法相比较于 NBI 算法在多样性方面的结果。

表6-2和表6-3分别给出了 NBI 和 USD 算法在 MovieLens 和 MovieTweetings 数据集中的推荐结果对比情况。通过比较可以发现，不管是在 MovieLens 还是 MovieTweetings 数据集上，USD 算法都可以大幅度的提升推荐结果的个性化指标，其中多样性、覆盖率和新颖性都显著的优于 NBI 算法。这说明 USD 算法可以推荐更多的长尾物品，为用户推荐的物品差异性也更大。而在推荐准确率方面，USD 算法在 MovieLens 数据集上的效果还要优于 NBI 算法。在 MovieTweetings 数据集中，USD 算法的准确率出现了小幅度的降低，比 NBI 算法低了 1.23%，但是相比较于推荐结果个性化水平的提高，这种降低还是值得的。

表 6-2 MovieLens 数据集上 USD 和 NBI 算法的推荐结果对比

| 算法 | 准确率 | 多样性 | 覆盖率 | 新颖性 |
|-----|---------------|---------------|---------------|---------------|
| USD | 0.1529 | 0.8737 | 0.0911 | 302.20 |
| NBI | 0.1459 | 0.7757 | 0.0630 | 346.36 |

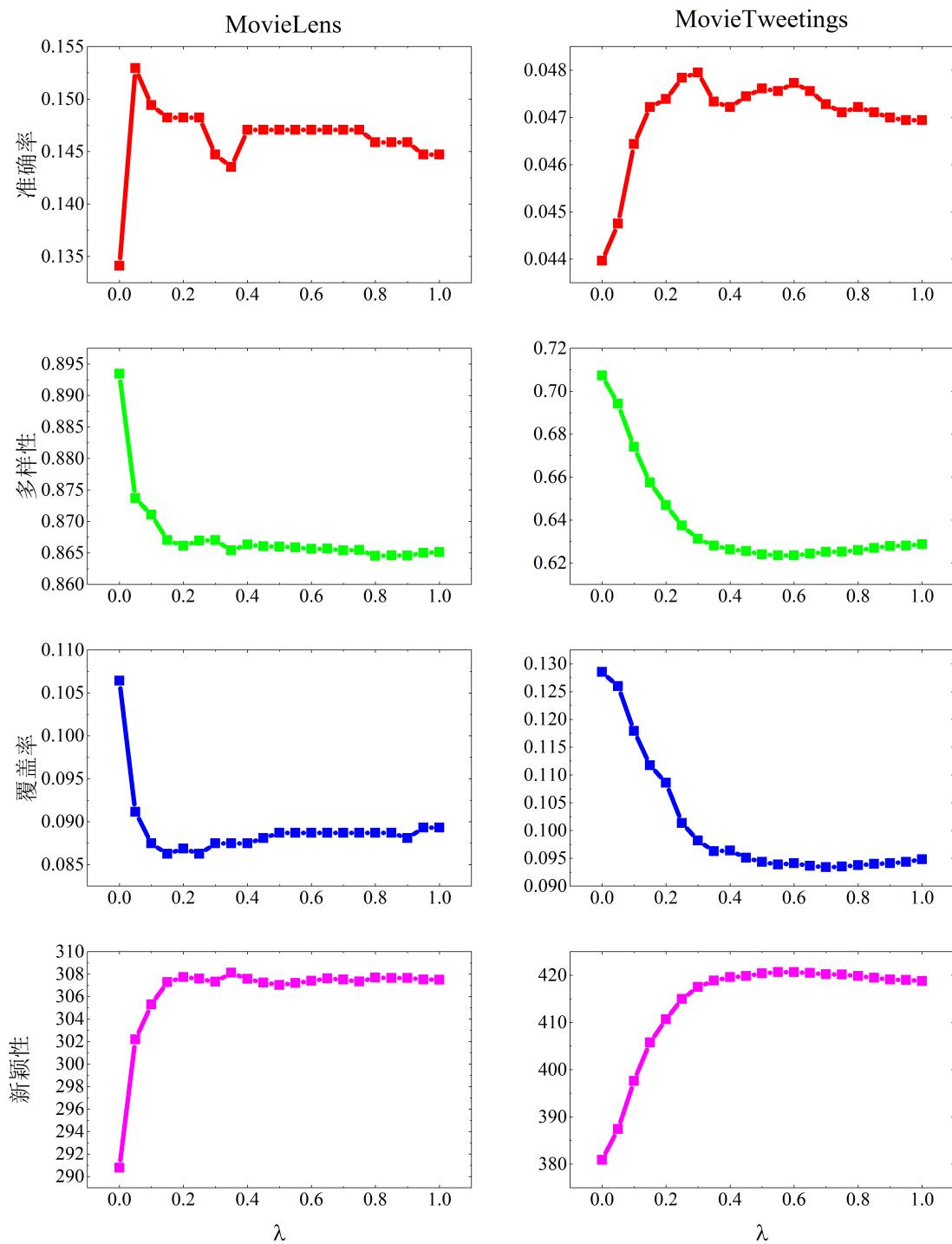


图 6-3 USD 算法中参数对推荐性能的影响

表 6-3 MovieTweetings 数据集上 USD 和 NBI 算法的推荐结果对比

| 数据集 | 准确率 | 多样性 | 覆盖率 | 新颖性 |
|-----|---------------|---------------|---------------|---------------|
| USD | 0.0479 | 0.6312 | 0.0982 | 417.51 |
| NBI | 0.0485 | 0.4916 | 0.0385 | 466.27 |

图6-4给出了在 MovieLens 和 MovieTweetings 数据集中，相较于 NBI 算法倾向推荐度超过 100 的物品，USD 算法推荐了更多度在 20 到 50 之间的物品，从而提供了很好的新颖性指标。在 MovieTweetings 数据集中，USD 算法更多地推荐了度小于 50 的物品，推荐列表中大度物品和小度物品出现的比例更加平衡，这也是 MovieTweetings 数据集中覆盖率大幅提升的原因。

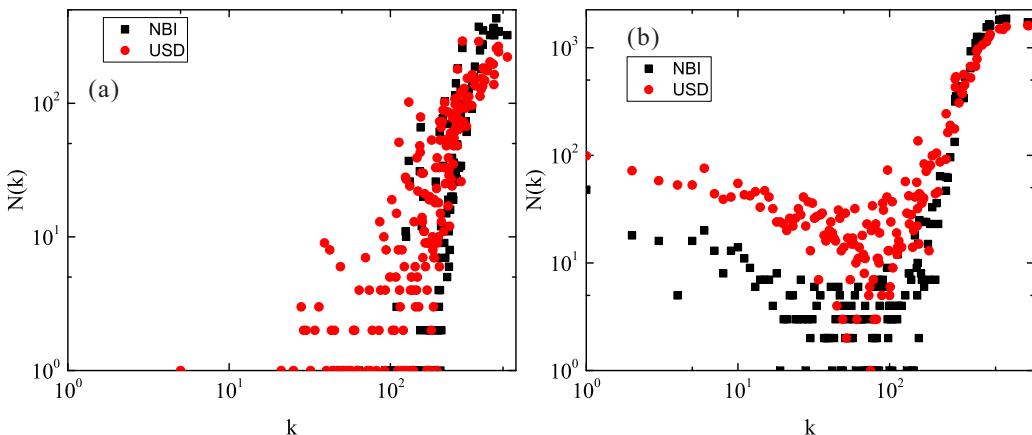


图 6-4 NBI 和 USD 算法推荐列表总物品的度分布。(a) Movielens; (b) MovieTweetings

6.4.3 基于物品序列的资源分配算法性能分析

ISD 算法使用了 Top-k 概率模型来计算用户相似度。不同的 k 值可能会对推荐结果产生不同的影响。因为 k 值的增大会保留更多的共同购买物品数，越大的 k 值通常情况下会得到更高的准确率；另一方面，更高的 k 值会使得计算复杂度成指数级增长。为了分析 k 的选取对 ISD 算法推荐性能的影响，本章在 MovieTweetings 数据集中对 $k = 1, 2, 3$ 和 4 时的四个推荐指标进行了比较分析，结果显示在表6-4中。

从表6-4中可以发现，当 k 逐渐增大时，ISD 算法的推荐准确率、多样性、覆盖率和新颖性指标的确都在提高，推荐性能变得越来越好。但是这种推荐性能的提升相比于计算耗时并不明显。例如，当 $k = 4$ 时的准确率比 $k = 1$ 的准确率提高了 1.32%，但整个推荐过程多耗费了一个小时的时间。由此可以发现 k 值的增加可

能显得得不偿失。在下文的对比实验中 k 值被设置为 4。

图6-5给出了在 MovieLens 和 MovieTweetings 数据集上 NBI 和 ISD 算法推荐的物品分布情况。在 MovieLens 数据集中，相较于 NBI 算法，ISD 算法并没有明显改变被推荐物品的度分别情况。但在 MovieTweetings 数据集中，可以发现被推荐物品的度分布整体向左上偏移了，说明更多的小度物品被推荐给了用户。ISD 和 NBI 算法的具体性能对比将在下节中介绍。

表 6-4 参数 k 对基于物品序列的资源分配算法的影响

| 评价指标 | K=1 | K=2 | K=3 | K=4 |
|------|--------|--------|--------|---------------|
| 准确率 | 0.0455 | 0.0459 | 0.0460 | 0.0461 |
| 覆盖率 | 0.1043 | 0.1030 | 0.1040 | 0.1047 |
| 多样性 | 0.6745 | 0.6766 | 0.6777 | 0.6778 |
| 新颖性 | 396.94 | 395.38 | 394.56 | 394.26 |
| 运行时间 | 57s | 74s | 334s | 3895s |

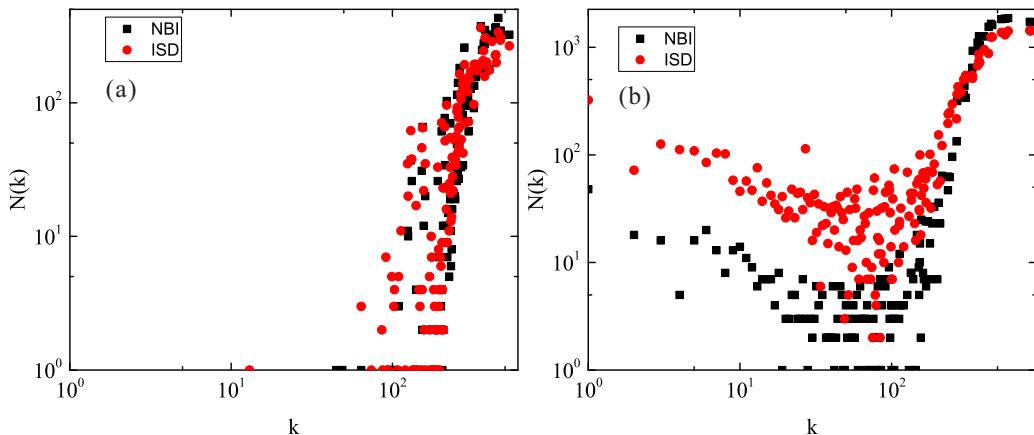


图 6-5 NBI 和 ISD 算法推荐列表总物品的度分布。(a) Movielens; (b) MovieTweetings

6.4.4 不同算法的推荐性能对比

表6-5和表6-6分别显示了 ISD、USD、NBI、HPH 和 BHC 算法在 MovieLens 和 MovieTweetings 数据集上的推荐结果对比情况。因为 USD 算法已经与 NBI 算法进行过了对比，因此首先将 ISD 算法与 NBI 算法进行对比分析。从推荐结果中可以发现 ISD 算法除了在 MovieTweetings 中的准确率有所降低，4 个评价指标在两个数据集中的推荐效果都要优于 NBI 算法。ISD 算法在 MovieLens 和 MovieTweetings 数据集上的覆盖率要比 NBI 算法分别提高了 18.41% 和 155.06%，而在新颖性方面，ISD 算法得到的结果都要低于 NBI 算法。这两个结果表明 ISD 算法相较于 NBI 算

法可以将系统中更多的商品推荐给用户，并且这些商品更加的小众，在一定程度上解决推荐系统的长尾问题和冷启动问题，这与 ISD 算法提供的推荐列表中物品的度分布情况吻合。多样性是用来衡量个性化程度的重要指标，表示用户与用户之间推荐列表的差异性。研究表明基于资源分配的推荐算法更加倾向于推荐大度的流行物品，从而使推荐给用户的物品比较集中，牺牲了多样性指标。从 MovieLens 和 MovieTweetings 两个数据集中的结果可以发现，ISD 算法可以为不同的用户产生更加个性化的推荐列表，多样性在两个数据集上分别提高了 7.5% 和 37.88%。当然，在分析推荐结果的个性化水平时，准确性依然是衡量一个推荐算法优劣的首要指标。通过比较两个数据集上的推荐准确率可以发现，ISD 算法在提高个性化指标的同时，并没有以牺牲准确率作为代价，仅比 NBI 算法在 MovieTweetings 数据集上降低了 4.9%，而在 MovieLens 数据集上 ISD 算法的推荐准确率要优于 NBI 算法。

表 6-5 MovieLens 数据集上各算法的推荐结果对比

| 评价指标 | NBI | USD | ISD | HPH | BHC |
|------|--------|---------------|---------------|--------|--------|
| 准确率 | 0.1459 | 0.1529 | 0.1588 | 0.1535 | 0.1483 |
| 覆盖率 | 0.0630 | 0.0911 | 0.0746 | 0.0786 | 0.0701 |
| 多样性 | 0.7757 | 0.8737 | 0.8339 | 0.8406 | 0.8381 |
| 新颖性 | 346.36 | 302.20 | 326.27 | 320.27 | 320.62 |
| 运行时间 | 14s | 26s | 57s | 43s | 34s |

表 6-6 MovieTweetings 数据集上各算法的推荐结果对比

| 评价指标 | NBI | USD | ISD | HPH | BHC |
|------|--------|--------|---------------|--------|---------------|
| 准确率 | 0.0485 | 0.0479 | 0.0461 | 0.0492 | 0.0512 |
| 覆盖率 | 0.0385 | 0.0982 | 0.1047 | 0.0537 | 0.0363 |
| 多样性 | 0.4916 | 0.6312 | 0.6778 | 0.5229 | 0.4613 |
| 新颖性 | 466.27 | 417.51 | 394.26 | 459.55 | 464.86 |
| 运行时间 | 12s | 27s | 19s | 31s | 25s |

相比于标准的 NBI 算法，USD 和 ISD 算法在保留较高推荐准确率的情况下，还能提供更加个性化的推荐结果。因此，本章将这两个算法与 HPH 和 BHC 算法进行了对比分析。HPH 和 BHC 算法都是基于热传导的改进推荐算法，能够提供既准确又多样的推荐结果。从表6-5和表6-6中可以发现，在 MovieLens 数据集中，USD 算法的个性化水平是最高的，ISD 算法的个性化水平与 BHC 算法基本持平，但要比 HPH 算法稍差；而在 MovieTweetings 数据集中，ISD 算法提供的推荐结果

个性化水平要优于 HPH 和 BHC 算法。这表明 ISD 算法利用全局的用户行为序列信息计算的用户间相似度能够一定程度上缓解数据稀疏的问题，在稀疏数据集上提供更加个性化的推荐结果。

6.5 本章小结

本章主要研究了在没有明确评分信息和具体时间信息的场景中，只根据购买同一物品的用户相对顺序和同一用户购买的物品相对顺序这两种相对时间信息，分别提出了基于用户序列和基于物品序列的资源分配算法。通过在两个电影评分数据集上的实验表明两种基于相对时间的资源分配算法可以在保留较高准确率的情况下，显著提升推荐系统的个性化水平，尤其是在稀疏数据集中。

第七章 总结与展望

7.1 研究工作总结

互联网和移动互联网的快速发展使得人们走进了线上与线下生活相互融合的时代，推荐系统作为连接生产者和消费者之间的桥梁，一方面主动为消费者推荐可能会感兴趣的商品，帮助消费者应对信息过载的问题，另一方面也在帮助信息和服务的生产者，将生产的内容主动推送到合适的消费者面前，增加企业产品的合理曝光度，帮助企业增加利润。不管是消费者还是生产者，他们的兴趣和需求都随着时间不断地发生着改变。尤其是在如今这个快速发展的时代，对时间感知的推荐算法的研究，不仅可以捕捉用户随时可能发生变化的兴趣，还可以综合考虑用户个体和用户之间的行为顺序的相关性。不管是从理论研究的角度还是实际工业应用的角度看，研究时间感知的推荐算法都有着重要的意义。本文的主要研究工作和创新点包括以下几个方面：

(1) 研究了用户历史正负面评价对未来兴趣的时间影响力。提出了一种基于正负面评价的时间感知推荐算法，其中不仅考虑了用户长短期兴趣的不同作用，还加入了正负面评价的不同时序影响。为了应对真实环境的计算效率问题，进一步提出了基于曲线拟合的时间感知推荐算法，利用线性回归方式对正面和负面评价的时间效应曲线进行了近似拟合，得出了正负面评价的不同时间衰减因子。通过对比相关算法，提出的两种基于正负面评价的推荐算法除了可以显著提高推荐准确性以外，还能提高在稀疏数据集中的个性化指标。

(2) 研究了用户个体行为的时间特性，分析了与用户整体行为和系统随机行为之间的差异。通过基于行为时间感知的分类方式，将用户分为引领型邻居和跟随型邻居，并分析了不同类型邻居用户的行为差异。通过考虑用户之间的时序影响和用户个体的兴趣漂移现象，提出了一种基于资源分配的时间感知推荐算法。实验结果证明该算法可以显著提高推荐结果的准确性和新颖性。

(3) 分析了系统中物品流行度表现出的时间特性，并提出了基于近期流行度的资源分配和热传导算法。实验结果表现，物品近期流行度可以帮助热传导算法在保留较高多样性的情况下，显著提高推荐准确性，克服过分专注小众物品的缺陷。同时，提出了一种基于近期流行度的后处理算法，将其与资源分配算法相结合，显著提升了推荐系统的准确性指标。

(4) 根据用户行为发生的相对顺序，提出了基于用户序列的资源分配算法和基于物品序列的资源分配算法。在基于用户序列的资源分配算法中，根据不同用户

选择同一个物品的先后顺序对用户之间的影响进行了量化，并将其作为二部图网络的连边权重。在基于物品序列的资源分配算法中，每个用户被表示为由她购买过的物品组成的有序概率分布向量，通过 Jensen-Shannon 散度计算的用户间相似度被作为了二部图网络的连边权重。实验结果表明两种算法都可以显著提高推荐结果的个性化水平，发现更多系统中用户喜欢的长尾商品。

7.2 未来工作展望

本文从用户和物品、显性反馈和隐性反馈、用户个体和用户群体等多个角度对系统表现出的时间特性进行了研究分析，并且取得了一些进展，但是仍存在许多不足和后续需要开展的工作。结合本文的工作内容，以下列出了一些需要进一步解决的问题：

(1) 本文使用的推荐算法评价指标是针对静态系统的，无法全面的衡量时间感知的推荐算法的动态推荐性能。如何设计针对动态推荐算法的评测指标是该领域需要探索的一个重要问题。例如在多样性方面，不同时间段表现出的推荐多样性是衡量动态推荐算法多样性的可用指标。

(2) 本文虽然提出了许多利用用户行为时间特性的推荐算法，但是这些方法普遍涉及到多个参数。在实验部分，参数的选定往往采用遍历寻优的方法，这虽然能够分析不同参数对推荐性能的影响，并揭示一些背后的相关原因。但是这种方式会使相关算法实际应用的价值降低，如何采用高效的参数估计方法提高时间感知推荐算法的实际应用价值，是该领域又一个需要解决的问题。

(3) 本文分别就显性反馈和隐性反馈，以及从用户和物品角度对用户行为表现出的动态时间特性进行了研究，并以此提出了不同的时间感知推荐算法。但是在真实系统中，这些时间特性是同时存在的。用户既表现出自身的动态特性，也表现出群体间的动态特性，这些不同的动力学的耦合使得整个系统变得更加复杂。因此，多种时序特征的相互影响也是时间感知的推荐算法值得深入研究的主题。

(4) 虽然本文研究了系统中用户表现出的时间动力学特性，并以此提出了相关的时间感知算法。但是这些算法仍然是针对系统当前时刻的全量数据进行计算的，如何将时间动力学特性应用到推荐算法的增量计算过程中，在局部更新的过程中融入时间信息，这是时间感知的推荐算法有待继续研究的问题。

致 谢

时间转眼即逝，经过半年多的时间，终于完成了我的博士毕业论文。在此要向一直关心和支持我的各位老师、朋友和家人表达诚挚的感谢，感谢你们为我的科研工作和平时生活提供的全面支持和帮助。

首先要感谢我的指导老师傅彦教授。在攻读硕士和博士学位期间，傅老师在学习和生活的各个方面都给予了我很大的帮助。不仅教会我如何脚踏实地的进行科研工作，还教会我如何将科研成果应用到实际工业领域。傅老师教会我不能只做一个在象牙塔中的读书人，还应该走出校园将自己所学习到的知识带入社会，为社会的进步和发展尽自己的一份力。感谢傅老师对我的精心栽培和关心。

感谢周俊临副教授，是周老师带我走入了科研的道路。第一次遇见周老师时，他还在攻读博士学位，那时的他已经是实验室的带头人，表现出出色的科研和工程能力。经由周老师的耐心指导，我对推荐系统领域产生了强烈兴趣，由此走上了科研学术的道路。从认识周老师开始到现在，他一直在科研工作和生活中关心我，为我付出了很多心血。

感谢董强副教授，在生活和科学研究方面给予了我很多帮助。读博期间，董老师经常与我一起讨论学术问题，耐心指导我的科研工作，教会我如何从细节出发做好每一件事情。感谢董老师对我毕业论文从选题开始到结束的一直支持。

感谢互联网科学中心提供给我的成长平台。从 2010 年进入实验室攻读硕士起，与实验室的老师和同学们已经一起走过了 7 年时间。在这期间，大家一起奋斗一起进步，见证了实验室的慢慢壮大。感谢实验室中的高辉老师、陈端兵老师、刘震老师在我读博期间的帮助和指导，感谢他们平时对我耐心的解惑。感谢已经毕业的崔爱香博士、聂大成博士、孙崇敬博士、赵志丹博士，他们不仅在学术研究方面对我进行了耐心的指导，更在平时的生活中给予了我很多帮助。感谢还未毕业的安亚会博士、李艳丽博士、费清菁博士、王超博士，平时与他们对学术的探讨和交流使我有了很大的进步。当然也要感谢正在攻读硕士学位的师扬波、李诚铭学弟，正因为有了你们的帮助才使我顺利完成了相关的科学实验工作。同时还要感谢已经毕业参加工作的白建国、冉秋萍、韦思炜、龚诗楠、定明静、虞浩济，感谢你们在学术研究与工业应用的融合方面对我的帮助和指导。

此外，还要感谢我的父母和家人。在读博期间，父母和家人一直在默默的支持我，是我背后最大的动力。在撰写毕业论文期间，母亲癌症复发，但她依然每天关心我的学习和生活，默默忍受着病魔的煎熬，坚强的生活着。感谢父亲对母亲

无微不至的照顾，替我承担了作为儿子的责任，无法在身边尽孝的我只想对你们说句“我爱你们”。感谢我的女朋友，感谢你在我读博期间对我的理解与支持，感谢你在我无法照顾父母时对我父母的关心与照顾。感谢所有关心和支持我的朋友们，谢谢你们为我无私的付出。

最后，感谢在百忙之中抽出时间评阅我博士学位论文的各位专家，谢谢你们！

参考文献

- [1] R. Yadav, S. K. Sharma, A. Tarhini. A multi-analytical approach to understand and predict the mobile commerce adoption[J]. Journal of Enterprise Information Management, 2016, 29(2): 222-237.
- [2] J. Schor. Debating the sharing economy[J]. Journal of Self-Governance and Management Economics, 2016, 4(3): 7-22.
- [3] G. P. Fettweis. The tactile internet: Applications and challenges[J]. IEEE Vehicular Technology Magazine, 2014, 9(1): 64-70.
- [4] 孙铁利, 邓凯英. 基于 web 的文本信息检索技术 [J]. 信息技术, 2007, 9: 127-129.
- [5] 陈佳宁. 移动音频类应用对传统广播的冲击与启示 [J]. 中国广播电视台学刊, 2015, 6: 56-60.
- [6] 丁正洪, 岳才勇. “微”言大义“博”精彩——政务微博运营要点 [J]. 中国公路, 2014, 5: 51-53.
- [7] J. Chevalier, A. Goolsbee. Measuring prices and price competition online: Amazon.com and barnesandnoble.com[J]. Quantitative Marketing and Economics, 2003, 1(2): 203-222.
- [8] 孙龙. 基于淘宝开放平台的“正品”排序策略的研究 [D]. 内蒙古: 内蒙古科技大学, 2013, 2-4.
- [9] P. Maes. Agents that reduce work and information overload[J]. Communications of the ACM, 1995, 37(7): 811-821.
- [10] A. Edmunds, A. Morris. The problem of information overload in business organisations: a review of the literature[J]. International Journal of Information Management, 2000, 20(1): 17-28.
- [11] M. Morita, Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval[C]. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 1994, 272-281.
- [12] J. Bobadilla, F. Ortega, A. Hernando, et al. Recommender systems survey[J]. Knowledge-Based Systems, 2013, 46(1): 109-132.
- [13] L. Y. Lü, M. Medo, H. Y. Chi, et al. Recommender systems[J]. Physics Reports, 2012, 519(1): 1-49.
- [14] R. Burke. Hybrid recommender systems: Survey and experiments[J]. User Modeling and User-Adapted Interaction, 2002, 12(4): 331-370.

- [15] 郑重. 21 天搭建推荐系统：实现“千人千面”个性化推荐 [EB/OL]. <https://yq.aliyun.com/articles/39629>, 2016-05-06.
- [16] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems[J]. Data Mining and Knowledge Discovery, 2012, 24(3): 555-583.
- [17] J. Bennett, S. Lanning, et al. The netflix prize[C]. Proceedings of KDD Cup and Workshop, San Jose, California, 2007, 35.
- [18] G. Linden, B. Smith, J. York. Amazon.com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76-80.
- [19] J. He. A social network-based recommender system[D]. Los Angeles: University of California, 2010, 47-74.
- [20] M. Bressan, S. Leucci, A. Panconesi, et al. The limits of popularity-based recommendations, and the role of social ties[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016, 745-754.
- [21] Z. Zhang, H. Lin, K. Liu, et al. A hybrid fuzzy-based personalized recommender system for telecom products/services[J]. Information Sciences, 2013, 235(6): 117-129.
- [22] J. Lu, Q. Shambour, Y. Xu, et al. A web-based personalized business partner recommendation system using fuzzy semantic techniques[J]. Computational Intelligence, 2013, 29(29): 37-69.
- [23] G. Adomavicius, A. Tuzhilin. Context-aware recommender systems[J]. AI Magazine, 2010, 32(3): 217-253.
- [24] Y. L. Chen, L. C. Cheng, C. N. Chuang. A group recommendation system with consideration of interactions among group members[J]. Expert Systems with Applications, 2008, 34(3): 2082-2090.
- [25] J. Masthoff. Group recommender systems: Combining individual models[M]. Boston: Springer US, 2011, 677-702.
- [26] A. L. Barabási. The architecture of complexity[J]. Proceedings of the American Philosophical Society, 2007, 27(4): 33-42.
- [27] T. Zhou, X. P. Han, B. H. Wang. Science matters: Humanities as complex systems[M]. Singapore City: World Scientific, 2008, 207-233.
- [28] A. L. Barabási. Bursts:the hidden patterns behind everything we do, from your e-mail to bloody crusades[M]. New York: Penguin, 2010.

- [29] C. J. Zhang, A. Zeng. Behavior patterns of online users and the effect on information filtering[J]. *Physica A: Statistical Mechanics and Its Applications*, 2012, 391(4): 1822-1830.
- [30] M. Shang, L. Lu, Y. C. Zhang, et al. Empirical analysis of web-based user-object bipartite networks[J]. *Europhysics Letters*, 2010, 90(4): 1303-1324.
- [31] Y. Liu, C. Liu, B. Liu, et al. Unified point-of-interest recommendation with temporal interval assessment[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016, 1015-1024.
- [32] H. Li, Y. Ge, R. Hong, et al. Point-of-interest recommendations: Learning potential check-ins from friends[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 2016, 975-984.
- [33] T. H. Dao, S. R. Jeong, H. Ahn. A novel recommendation model of location-based advertising: Context-aware collaborative filtering using ga approach[J]. *Expert Systems with Applications*, 2012, 39(3): 3731-3739.
- [34] S. Scellato, A. Noulas, C. Mascolo. Exploiting place features in link prediction on location-based social networks[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA, 2011, 1046-1054.
- [35] A. L. Barabási. The origin of bursts and heavy tails in human dynamics[J]. *Nature*, 2005, 435(7039): 207.
- [36] P. G. Campos, F. Díez, A. Bellogín. Temporal rating habits: a valuable tool for rating discrimination[C]. Challenge on Context-Aware Movie Recommendation, Chicago, USA, 2011, 29-35.
- [37] L. Baltrunas, X. Amatriain. Towards time-dependant recommendation based on implicit feedback[C]. Workshop on Context-aware Recommender Systems, New York, NY, USA, 2009.
- [38] U. Panniello, A. Tuzhilin, M. Gorgoglione, et al. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems[C]. ACM Conference on Recommender Systems, New York, NY, USA, 2009, 265-268.
- [39] Y. Koren. Collaborative filtering with temporal dynamics[J]. *Communications of the ACM*, 2010, 53(4): 89-97.
- [40] J. Liu, G. Deng. Link prediction in a user-object network based on time-weighted resource allocation[J]. *Physica A: Statistical Mechanics and Its Applications*, 2009, 388(17): 3643-3650.
- [41] C. Castellano, S. Fortunato, V. Loreto. Statistical physics of social dynamics[J]. *Review of Modern Physics*, 2007, 81(2): 591-646.

- [42] G. Roger, L. Alejandro, M. Esteban, et al. Predicting human preferences using the block structure of complex social networks[J]. Plos One, 2012, 7(9): e44620.
- [43] D. Lazer, A. S. Pentland, L. Adamic, et al. Life in the network: The coming age of computational social science[J]. Science, 2016, 323(5915): 721-723.
- [44] J. G. Oliveira, A. L. Barabási. Human dynamics: Darwin and einstein correspondence patterns[J]. Nature, 2005, 437(7063): 1251.
- [45] D. Z, A. E, L. A, et al. Dynamics of information access on the web[J]. Physical Review E, 2006, 73(6): 066132.
- [46] T. Zhou, H. A. T. Kiet, B. J. Kim, et al. Role of activity in human dynamics[J]. Europhysics Letters, 2008, 82(2): 28002.
- [47] B. Gonçalves, J. J. Ramasco. Human dynamics revealed through web analytics[J]. Physical Review E, 2008, 78(2): 026123.
- [48] Y. Wu, H. J. Schellnhuber. Evidence for a bimodal distribution in human communication[J]. Proceedings of the National Academy of Sciences of the United States of America, 2010, 107(44): 18803.
- [49] A. Vázquez, J. G. Oliveira, Z. Dezsö, et al. Modeling bursts and heavy tails in human dynamics[J]. Physical Review E, 2006, 73(2): 036127.
- [50] R. D. Malmgren, D. B. Stouffer, A. E. Motter, et al. A poissonian explanation for heavy tails in e-mail communication[J]. Proceedings of the National Academy of Sciences of the United States of America, 2008, 105(47): 18153-8.
- [51] R. D. Malmgren, D. B. Stouffer, A. S. L. O. Campanharo, et al. On universality in human correspondence activity[J]. Science, 2009, 325(5948): 1696-1700.
- [52] X. P. Han, T. Zhou, B. H. Wang. Modeling human dynamics with adaptive interest[J]. New Journal of Physics, 2007, 10(7): 1983-1989.
- [53] A. Vazquez. Impact of memory on human dynamics[J]. Physica A: Statistical Mechanics and Its Applications, 2006, 373(36): 747-752.
- [54] J. G. Oliveira, A. Vazquez. Impact of interactions on human dynamics[J]. Physica A: Statistical Mechanics and Its Applications, 2009, 388(2): 187-192.
- [55] B. Min, K. I. Goh, I. M. Kim. Waiting time dynamics of priority-queue networks[J]. Physical Review E, 2009, 79(5): 056110.

- [56] Q. Yuan, G. Cong, Z. Ma, et al. Time-aware point-of-interest recommendation[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 2013, 363-372.
- [57] J. D. Zhang, C. Y. Chow. Ticrec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations[J]. IEEE Transactions on Services Computing, 2016, 9(4): 633-646.
- [58] Z. D. Zhao, Z. Yang, Z. Zhang, et al. Emergence of scaling in human-interest dynamics[J]. Science Report, 2013, 3(12): 3472.
- [59] Z. Yang, Z.-K. Zhang, T. Zhou. Anchoring bias in online voting[J]. Europhysics Letters, 2013, 100(6): 68002.
- [60] B. P. S. Murthi, S. Sarkar. The role of the management sciences in research on personalization[J]. Management Science, 2003, 49(10): 1344-1362.
- [61] J. B. Schafer, J. Konstan, J. Riedl. Recommender systems in e-commerce[C]. ACM Conference on Electronic Commerce, Denver, Colorado, USA, 1999, 158-166.
- [62] Q. L. Tong, Y. Park, Y. T. Park. A time-based approach to effective recommender systems using implicit feedback[J]. Expert Systems with Applications, 2008, 34(4): 3055-3062.
- [63] G. Adomavicius, A. Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(17): 734-749.
- [64] J. B. Schafer, F. Dan, J. Herlocker, et al. Collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2011, 4(2): 81-173.
- [65] M. J. Pazzani, D. Billsus. The adaptive web[M]. Boston: Springer US, 2007, 325-341.
- [66] R. Burke. Knowledge-based recommender systems[J]. Encyclopedia of Library and Information Systems, 2000, 69: 175-186.
- [67] R. B. Allen. User models: Theory, method, practice[J]. International Journal of Man-Machine Studies, 1990, 32(5): 511-543.
- [68] S. Lee, J. Yang, S. Y. Park. Discovery of hidden similarity on collaborative filtering to overcome sparsity problem[C]. International Conference on Discovery Science, Padova, Italy, 2004, 396-402.
- [69] P. Resnick, N. Iacovou, M. Suchak, et al. GroupLens: an open architecture for collaborative filtering of netnews[C]. ACM Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, 1994, 175-186.

- [70] B. Sarwar, G. Karypis, J. Konstan, et al. Item-based collaborative filtering recommendation algorithms[C]. International Conference on World Wide Web, Hong Kong, China, 2001, 285-295.
- [71] M. Deshpande, G. Karypis. Item-based top- n recommendation algorithms[J]. ACM Transactions on Information Systems, 2004, 22(1): 143-177.
- [72] M. A. Ghazanfar, A. Gel-Bennett, S. Szemek. Kernel-mapping recommender system algorithms[J]. Information Sciences, 2012, 208(21): 81-104.
- [73] I. Rish. An empirical study of the naive bayes classifier[J]. Journal of Universal Computer Science, 2001, 1(2): 127.
- [74] Y. Koren, R. Bell, C. Volinsky. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [75] X. He, H. Zhang, M.-Y. Kan, et al. Fast matrix factorization for online recommendation with implicit feedback[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, 2016, 549-558.
- [76] Y. B. Cho, Y. H. Cho, S. H. Kim. Mining changes in customer buying behavior for collaborative recommendations[J]. Expert Systems with Applications, 2005, 28(2): 359-369.
- [77] T. Zhou, J. Ren, M. Medo, et al. Bipartite network projection and personal recommendation[J]. Physical Review E, 2007, 76(2): 046115.
- [78] Y. C. Zhang, M. Medo, J. Ren, et al. Recommendation model based on opinion diffusion[J]. Europhysics Letters, 2007, 80(6): 417-429.
- [79] Y. C. Zhang, M. Blattner, Y. K. Yu. Heat conduction process on community networks as a recommendation model[J]. Physical Review Letters, 2007, 99(15): 154301.
- [80] T. Zhou, G. Parisi. Solving the apparent diversity-accuracy dilemma of recommender systems[J]. Proceedings of the National Academy of Sciences of the United States of America, 2010, 107(10): 4511.
- [81] J. G. Liu, T. Zhou, Q. Guo. Information filtering via biased heat conduction[J]. Physical Review E, 2011, 84(2): 037101.
- [82] Z. K. Zhang, C. Liu, Y. C. Zhang, et al. Solving the cold-start problem in recommender systems with social tags[J]. Europhysics Letters, 2010, 92(2): 28002.
- [83] W. Yao, J. He, G. Huang, et al. Personalized recommendation on multi-layer context graph[J]. Lecture Notes in Computer Science, 2013, 8180: 135-148.
- [84] Z. Lu, D. Agarwal, I. S. Dhillon. A spatio-temporal approach to collaborative filtering[C]. ACM Conference on Recommender Systems, New York, NY, USA, 2009, 13-20.

- [85] A. Zimdars, D. M. Chickering, C. Meek. Using temporal data for making recommendations[C]. Conference on Uncertainty in Artificial Intelligence, Seattle, Washington, 2001, 580-588.
- [86] T. Y. Tang, P. Winoto, K. C. Chan. Scaling down candidate sets based on the temporal feature of items for improved hybrid recommendations[C]. International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2005, 169-186.
- [87] Y. Ding, X. Li, M. E. Orlowska. Recency-based collaborative filtering[C]. Australasian Database Conference, Hobart, Australia, 2006, 99-107.
- [88] T. Nathanson, E. Bitton, K. Goldberg. Eigentaste 5.0: constant-time adaptability in a recommender system using item clustering[C]. ACM Conference on Recommender Systems, Minneapolis, MN, USA, 2007, 149-152.
- [89] L. Xiang, Q. Yang. Time-dependent models in collaborative filtering based recommender system[C]. IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, Milan, Italy, 2009, 450-457.
- [90] N. Lathia, S. Hailes, L. Capra. Temporal collaborative filtering with adaptive neighbourhoods[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval, Boston, Massachusetts, USA, 2009, 796-797.
- [91] L. Xiang, Q. Yuan, S. Zhao, et al. Temporal recommendation on graphs via long- and short-term preference fusion[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2010, 723-732.
- [92] W. J. Song, Q. Guo, J. G. Liu. Improved hybrid information filtering based on limited time window[J]. Physica A: Statistical Mechanics and Its Applications, 2014, 416(416): 192-197.
- [93] T. De Pessemier, S. Dooms, T. Deryckere, et al. Time dependency of data quality for collaborative filtering algorithms[C]. ACM Conference on Recommender Systems, Barcelona, Spain, 2010, 281-284.
- [94] T. Jambor, J. Wang, N. Lathia. Using control theory for stable and efficient recommender systems[C]. International Conference on World Wide Web, New York, NY, USA, 2012, 11-20.
- [95] Y. Ding, X. Li. Time weight collaborative filtering[C]. ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, 2005, 485-492.
- [96] Y. Ren, T. Zhu, G. Li, et al. Top-n recommendations by learning user preference dynamics[C]. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Gold Coast, Australia, 2013, 390-401.

- [97] C. L. Huang, W. L. Huang. Handling sequential pattern decay: Developing a two-stage collaborative recommender system[J]. *Electronic Commerce Research and Applications*, 2009, 8(3): 117-129.
- [98] T. Chen, W. L. Han, H. D. Wang, et al. Content recommendation system based on private dynamic user profile[C]. *International Conference on Machine Learning and Cybernetics*, Hong Kong, China, 2007, 2112-2118.
- [99] N. Lathia, S. Hailes, L. Capra, et al. Temporal diversity in recommender systems[C]. *International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland, 2010, 210-217.
- [100] D. H. Park, H. K. Kim, I. Y. Choi, et al. A literature review and classification of recommender systems research[J]. *Expert Systems with Applications*, 2012, 39(11): 10059-10072.
- [101] C.-M. Chen, M.-F. Tsai, Y.-C. Lin, et al. Query-based music recommendations via preference embedding[C]. *ACM Conference on Recommender Systems*, Boston, Massachusetts, USA, 2016, 79-82.
- [102] D. Billsus, M. J. Pazzani. User modeling for adaptive news access[J]. *User Modeling and User-Adapted Interaction*, 2000, 10(2): 147-180.
- [103] O. R. Zaíane. Building a recommender agent for e-learning systems[C]. *International Conference on Computers in Education*, Auckland, New Zealand, 2002, 55-59.
- [104] J. Lu. Personalized e-learning material recommender system[C]. *International Conference on Information Technology for Application*, Harbin, China, 2004, 374-379.
- [105] X. N. Lam, T. Vu, T. D. Le, et al. Addressing cold-start problem in recommendation systems[C]. *International Conference on Ubiquitous Information Management and Communication*, Suwon, Korea, 2008, 208-211.
- [106] M. Aharon, O. Anava, N. Avigdor-Elgrabli, et al. Excuseme: Asking users to help in item cold-start recommendations[C]. *ACM Conference on Recommender Systems*, Vienna, Austria, 2015, 83-90.
- [107] Z. Huang, H. Chen, D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering[J]. *ACM Transactions on Information Systems*, 2004, 22(1): 116-142.
- [108] T. Zhou, 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.

- [109] J. G. Liu, T. Zhou, H. A. Che, et al. Effects of high-order correlations on personalized recommendations for bipartite networks[J]. *Physica A: Statistical Mechanics and Its Applications*, 2010, 389(4): 881-886.
- [110] J. S. Breese, D. Heckerman, C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering[C]. Conference on Uncertainty in Artificial Intelligence, San Francisco, California, USA, 1998, 43-52.
- [111] W. Y. Chen, J. C. Chu, J. Luan, et al. Collaborative filtering for orkut communities: discovery of user latent behavior[C]. International Conference on World Wide Web, New York, NY, USA, 2009, 681-690.
- [112] R. Gemulla, E. Nijkamp, P. J. Haas, et al. Large-scale matrix factorization with distributed stochastic gradient descent[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, 569-577.
- [113] B. Sarwar, G. Karypis, J. Konstan, et al. Incremental singular value decomposition algorithms for highly scalable recommender systems[C]. International Conference on Computer and Information Science, 2002, 27-28.
- [114] C. N. Ziegler, S. M. McNee, J. A. Konstan, et al. Improving recommendation lists through topic diversification[C]. International Conference on World Wide Web, New York, NY, USA, 2005, 22-32.
- [115] N. Hurley, M. Zhang. Novelty and diversity in top-n recommendation-analysis and evaluation[J]. *ACM Transactions on Internet Technology*, 2011, 10(2011): 14.
- [116] B. Mobasher, R. Burke, R. Bhaumik, et al. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness[J]. *ACM Transactions on Internet Technology*, 2007, 7(4): 23.
- [117] S. K. Lam, Tony, D. Frankowski, et al. Do you trust your recommendations? an exploration of security and privacy issues in recommender systems[C]. International Conference on Emerging Trends in Information and Communication Security, Freiburg, Germany, 2006, 14-29.
- [118] P. Resnick, R. Sami. The influence limiter: provably manipulation-resistant recommender systems[C]. ACM Conference on Recommender Systems, Minneapolis, MN, USA, 2007, 25-32.
- [119] H. Yu, C. Shi, M. Kaminsky, et al. Dsybil: Optimal sybil-resistance for recommendation systems[C]. IEEE Symposium on Security and Privacy, Oakland, California, 2009, 283-298.
- [120] N. Ramakrishnan, B. J. Keller, B. J. Mirza, et al. Privacy risks in recommender systems[J]. *IEEE Internet Computing*, 2001, 5(6): 54.

- [121] E. Aïmeur, G. Brassard, J. M. Fernandez, et al. Alambic: a privacy-preserving recommender system for electronic commerce[J]. International Journal of Information Security, 2008, 7(5): 307-334.
- [122] H. Polat, W. Du. Svd-based collaborative filtering with privacy[C]. ACM Symposium on Applied Computing, Santa Fe, New Mexico, 2005, 791-795.
- [123] P. Melville, R. J. Mooney, R. Nagarajan. Content-boosted collaborative filtering for improved recommendations[C]. Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, 2002, 187-192.
- [124] S. K. Lee, Y. H. Cho, S. H. Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations[J]. Information Sciences, 2010, 180(11): 2142-2155.
- [125] D. Billsus, M. J. Pazzani. Learning collaborative information filters[C]. International Conference on Machine Learning, Madison, Wisconsin, USA, 1998, 46-54.
- [126] X. Su, T. M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms[C]. IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, USA, 2006, 497-504.
- [127] K. Miyahara, M. J. Pazzani. Collaborative filtering with the simple bayesian classifier[J]. Information Processing Society of Japan, 2000, 1886: 679-689.
- [128] B. M. Sarwar, G. Karypis, J. Konstan, et al. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering[C]. International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2002.
- [129] T. Hofmann, J. Puzicha. Latent class models for collaborative filtering[C]. Sixteenth International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999, 688-693.
- [130] D. Lemire, A. Maclachlan. Slope one predictors for online rating-based collaborative filtering[C]. SIAM International Conference on Data Mining, Newport Beach, California, 2005, 471-475.
- [131] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, 2008, 426-434.
- [132] Y. Ren, G. Li, W. Zhou. Learning rating patterns for top-n recommendations[C]. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, 2012, 472-479.

- [133] Y. Ren, G. Li, W. Zhou. A learning method for top-n recommendations with incomplete data[J]. Social Network Analysis and Mining, 2013, 3(4): 1135-1148.
- [134] L. Baltrunas, F. Ricci. Context-based splitting of item ratings in collaborative filtering[C]. ACM Conference on Recommender Systems, New York, NY, USA, 2009, 245-248.
- [135] T. Zhou, L. L. Jiang, R. Q. Su, et al. Effect of initial configuration on network-based recommendation[J]. Europhysics Letters, 2008, 81(5): 58004.
- [136] F. Ricci, L. Rokach, B. Shapira, et al. Recommender systems handbook[M]. Boston: Springer US, 2011, 1-35.
- [137] G. Ling, M. R. Lyu, I. King. Ratings meet reviews, a combined approach to recommend[C]. ACM Conference on Recommender Systems, Foster City, Silicon Valley, California, USA, 2014, 105-112.
- [138] P. Gao, H. Miao, J. S. Baras, et al. Star: Semiring trust inference for trust-aware social recommenders[C]. ACM Conference on Recommender Systems, Boston, Massachusetts, USA, 2016, 301-308.
- [139] A. Salehi-Abari, C. Boutilier. Preference-oriented social networks: Group recommendation and inference[C]. ACM Conference on Recommender Systems, Vienna, Austria, 2015, 35-42.
- [140] A. J. Chaney, D. M. Blei, T. Eliassi-Rad. A probabilistic model for using social networks in personalized item recommendation[C]. ACM Conference on Recommender Systems, Vienna, Austria, 2015, 43-50.
- [141] A. Gunawardana, C. Meek. A unified approach to building hybrid recommender systems[C]. ACM Conference on Recommender Systems, New York, NY, USA, 2009, 117-124.
- [142] M. Jahrer, A. Scher, R. Legenstein. Combining predictions for accurate recommender systems[C]. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 2010, 693-702.
- [143] J. L. Herlocker, J. A. Konstan, L. G. Terveen, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53.
- [144] 刘建国, 周涛, 汪秉宏. 个性化推荐系统评价方法综述 [J]. 复杂系统与复杂性科学, 2009, 6(3): 1-10.
- [145] 朱郁筱, 吕琳媛. 推荐系统评价指标综述 [J]. 电子科技大学学报, 2012, 41(2): 163-175.
- [146] S. Vargas, P. Castells. Improving sales diversity by recommending users to items[C]. ACM Conference on Recommender Systems, Foster City, Silicon Valley, California, USA, 2014, 145-152.

- [147] S. Dooms, T. D. Pessemier, L. Martens. Movietweetings: a movie rating dataset collected from twitter[C]. Workshop on Crowdsourcing and Human Computation for Recommender Systems, Hong Kong, China, 2013, 43.
- [148] W. Zeng, Y. X. Zhu, L. Lü, et al. Negative ratings play a positive role in information filtering[J]. *Physica A: Statistical Mechanics and Its Applications*, 2011, 390(23–24): 4486-4493.
- [149] J. G. Liu, B. H. Wang, Q. Guo. Improved collaborative filtering algorithm via information transformation[J]. *International Journal of Modern Physics C*, 2009, 20(02): 285-293.
- [150] Q. Guo, Y. Li, J. G. Liu. Information filtering based on users' negative opinions[J]. *International Journal of Modern Physics C*, 2013, 24(24): 50032.
- [151] J. G. Liu, Q. Guo, Y. C. Zhang. Information filtering via weighted heat conduction algorithm[J]. *Physica A: Statistical Mechanics and its Applications*, 2011, 390(12): 2414-2420.
- [152] M. Blattner, Y. C. Zhang, S. Maslov. Exploring an opinion network for taste prediction: An empirical study[J]. *Physica A: Statistical Mechanics and Its Applications*, 2007, 373(36): 753-758.
- [153] X. S. He, M. Y. Zhou, Z. Zhuo, et al. Predicting online ratings based on the opinion spreading process[J]. *Physica A: Statistical Mechanics and its Applications*, 2015, 436: 658-664.
- [154] E. Frolov, I. Oseledets. Fifty shades of ratings: How to benefit from a negative feedback in top-n recommendations tasks[C]. ACM Conference on Recommender Systems, Boston, Massachusetts, USA, 2016, 91-98.
- [155] J. L. Herlocker, J. A. Konstan, A. Borchers, et al. An algorithmic framework for performing collaborative filtering[C]. International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, USA, 1999, 230-237.
- [156] M. Tsagkias, W. Weerkamp, M. De Rijke. Predicting the volume of comments on online news stories[C]. ACM Conference on Information and Knowledge Management, Hong Kong, China, 2009, 1765-1768.
- [157] A. Zeng, S. Gualdi, M. Medo, et al. Trend prediction in temporal bipartite networks: the case of movielens, netflix, and digg[J]. *Advances in Complex Systems*, 2013, 16(04-05): 1350024.
- [158] P. D. B. Parolo, R. K. Pan, R. Ghosh, et al. Attention decay in science[J]. *Journal of Informetrics*, 2015, 9(4): 734-745.
- [159] P. Holme, J. Saramäki. Temporal networks[J]. *Physics Reports*, 2011, 519(3): 97-125.
- [160] M. Medo. Statistical validation of high-dimensional models of growing networks[J]. *Physical Review E*, 2014, 89(3): 032801.

- [161] Y. Zhou, L. Lü, W. Liu, et al. The power of ground user in recommender systems[J]. Plos One, 2013, 8(8): e70094.
- [162] X. Ren, L. Lü, R. Liu, et al. Avoiding congestion in recommender systems[J]. New Journal of Physics, 2014, 16(6): 063057.
- [163] T. Hofmann. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems, 2004, 22(1): 89-115.
- [164] T. Y. Liu. Learning to rank for information retrieval[J]. Foundations and Trends in Information Retrieval, 2009, 3(3): 225-331.

攻读博士学位期间取得的成果

- [1] W. J. Li, Y. Y. Xu, Q. Dong, et al. Tadb: A time-aware diffusion-based recommender algorithm[J]. International Journal of Modern Physics C, 2015, 26(9): 1550102. (SCI 小类学科“计算机：跨学科应用”，个人排名第一)
- [2] W. J. Li, Q. Dong, Y. B. Shi, et al. Effect of recent popularity on heat-conduction based recommendation models[J]. Physica A: Statistical Mechanics and its Applications, 2017, 474: 334-343. (SCI, 个人排名第一)
- [3] W. J. Li, Q. Dong, Y. Fu. Investigating the temporal effect of user preferences with application in movie recommendation[J]. Mobile Information Systems, 2017, 2017: 8940709. (SCI, 个人排名第一)
- [4] W. J. Li, Q. Dong, Y. Fu. List-wise diffusion-based recommender algorithm from implicit feedback[C]. International Conference on Information Science and Security, Seoul, Korea, 2015, 1-4. (EI 会议, 个人排名第一)
- [5] J. L. He, D. B. Chen, C. J. Sun, et al. Efficient stepwise detection of communities in temporal networks[J]. Physica A: Statistical Mechanics and its Applications, 2017, 469: 438-446. (SCI, 个人排名第五)

参加的科研项目

- [1] 项目负责人：四川省科技创新苗子工程《客运车辆智能监控系统》（项目编号：2012ZZ013）2012 年-2013 年；
- [2] 项目负责人：中央高校基本科研业务费（学生）项目《基于社交网络的简历推荐系统》2013 年-2014 年；
- [3] 参与：国家自然科学基金《基于社交网络的多样性小众推荐技术研究》（基金号：61103109）2012 年-2014 年；
- [4] 参与：国家自然科学基金《二进制递归网络的多故障容错性分析》（基金号：61300018）2014 年-2016 年；
- [5] 参与：国家自然科学基金《时变社交网络结构与传播动力学研究》（基金号：61673085）2017 年-2020 年。