摘　要

随着互联网的发展，大数据的到来，传统的音乐行业受到了很大的冲击，原有的音乐数字化给人们生活带来了极大的便利。随着数字音乐的兴起，各大音乐平台层出不穷，人们在音乐平台上收听音乐的时，常常因为歌曲信息繁杂，而不能找到自己想听的音乐。为了解决这个问题，音乐领域引入了推荐系统。本文在基于协同过滤的基础上，融合了基于内容的音乐推荐算法，并且设计了一个音乐推荐系统，主要内容如下：

（1）基于协同过滤的相似度改进。传统的基于协同过滤算法给用户推荐时，一些热门的歌曲，会影响用户与用户之间、歌曲与歌曲之间的相似度计算，导致推荐效果不佳，本文采用惩罚热门歌曲来降低热门歌曲对相似度的计算带来的影响，通过对相似度计算的改进，提高推荐的精度。

（2）基于协同过滤与内容的混合。在基于协同过滤的音乐推荐系统中，由于新用户在系统中没有用户的行为数据，那么系统很难为新用户推荐，即存在冷启动问题。针对该问题，本文提出了结合内容的推荐方法来解决。利用用户历史行为、用户标签信息，得到不同标签下的热门歌曲，将这些热门歌曲推荐给新用户，解决冷启动问题。并且在爬取的网易云音乐数据进行实验验证，证明了混合推荐算法的准确率和召回率优于基于协同过滤和内容的推荐算法。

（3）音乐推荐系统的设计与实现。在 Pycharm 平台上进行开发，系统前后端分离，前端采用 Vue.js 框架进行开发，后端采用 Python 的 Django 框架进行开发。设计实现了基于协同过滤的音乐推荐系统，满足用户个性化需求，最后以更友好的系统页面推荐歌曲给用户。

关键词：协同过滤；推荐系统；网络爬虫；冷启动

Abstract

With the development of the Internet and the arrival of big data, the traditional music industry has been greatly impacted by the digitization of the original music, which has brought great convenience to people's lives. With the rise of digital music, major music platforms have emerged, and when people listen to music on music platforms, they often cannot find the music they want to listen to because of the complicated song information. In order to solve this problem, recommendation systems have been introduced in the music field. This thesis incorporates a content-based music recommendation algorithm based on collaborative filtering, and designs a music recommendation system with the following main elements.

(1) Similarity improvement based on collaborative filtering. When the traditional collaborative filtering-based algorithm recommends to users, some popular songs, which will affect the similarity calculation between users and users and between songs and songs, lead to poor recommendation effect. In this thesis, we use penalized popular songs to reduce the calculation of similarity of popular songs, and improve the accuracy of recommendation by improving the similarity.

(2) Collaborative filtering and content based hybrid. In the collaborative filtering based music recommendation system, since new users do not have user's behavior data in the system, then it is difficult for the system to recommend for new users, i.e. there is a cold start problem. To address this problem, this thesis proposes a recommendation method combined with content to solve it. Using user's historical behavior and user tag information, we get the popular songs under different tags and recommend these popular songs to new users to solve the cold start problem. And the experimental validation on the crawled NetEase cloud music data proves that the accuracy and recall of the hybrid recommendation algorithm is better than the recommendation algorithm based on collaborative filtering and content.

(3) Design and implementation of music recommendation system. The system was developed on Pycharm platform, with the front and back ends separated, and the

front end was developed with Vue.js framework, and the back end was developed with Django framework in Python. We designed and implemented a collaborative filtering-based music recommendation system to meet users' personalized needs, and finally recommended songs to users with a more friendly system page.

Key Words： Collaborative filtering; Recommendation system; Web crawler; Cold start

目 录

摘 要 I

Abstract II

第 1 章 绪 论 1

1.1 研究背景与意义 1

1.2 国内外研究现状 3

1.2.1 音乐推荐系统研究现状 4

1.2.2 音乐推荐方法研究现状 5

1.3 本文的组织结构 6

1.4 本章小结 6

第 2 章 相关理论及技术介绍 8

2.1 网络爬虫技术 8

2.1.1 爬虫工作原理 8

2.1.2 爬虫的分类 8

2.2 常见的音乐推荐算法 8

2.2.1 基于协同过滤的推荐算法 9

2.2.2 基于内容的推荐算法 12

2.2.3 混合推荐算法 12

2.3 相似度的计算方法 14

2.3.1 皮尔逊相关系数 14

2.3.2 杰卡德相似度 14

2.3.3 余弦相似度 15

2.4 系统开发技术 15

2.4.1 前端相关技术 15

2.4.2 后端相关技术 15
2.5 本章小结 16
第 3 章 基于协同过滤和内容的音乐推荐算法 18
3.1 基于用户协同的音乐推荐算法 18
3.1.1 构建用户-歌曲偏好矩阵 19
3.1.2 改进用户相似度的计算 19
3.1.3 预测歌曲评分并推荐 21
3.2 基于物品协同的音乐推荐算法 21
3.2.1 构建用户-歌手偏好矩阵 21
3.2.2 改进歌手相似度的计算 22
3.2.3 预测歌手评分并推荐 23
3.3 基于内容的音乐推荐算法 24
3.3.1 音乐内容属性分析 24
3.3.2 对歌曲、歌单分别建模 25
3.3.3 预测歌曲、歌单评分并推荐 25
3.4 基于内容和协同过滤的混合推荐算法 27
3.4.1 混合推荐策略与流程 27
3.4.2 冷启动的解决 28
3.5 实验与分析 28
3.5.1 实验数据 28
3.5.2 实验评价指标 28
3.5.3 结果与分析 29
3.6 本章小结 30
第 4 章 音乐推荐系统的设计与实现 31
4.1 系统分析 31
4.1.1 可行性分析 31

4.1.2 功能性需求分析 32
4.1.3 非功能性需求分析 33
4.2 系统总体设计 33
4.2.1 功能模块设计 33
4.2.2 数据库设计 34
4.3 系统的关键功能模块实现 39
4.3.1 登录注册 39
4.3.2 为你推荐 41
4.3.3 歌单推荐 41
4.3.4 歌曲推荐 42
4.3.5 歌手推荐 43
4.3.6 推荐排行榜 43
4.3.7 其他模块 44
4.4 系统测试 45
4.5 本章小结 46
结　语 47
参考文献 48
致　谢 52
攻读硕士学位期间所取得的科研成果 53

第 1 章 绪　论

1.1 研究背景与意义

如今，互联网、信息技术的兴起，大数据、人工智能、5G 等相关技术的普及和应用，正在改变着人们的日常生活方式。由中国互联网信息中心发布的第 49 次《中国互联网络发展状况统计报告》[1]，此报告指出，据 2021 年 12 月统计的数据，中国网民已经达到 10.33 亿，与 2020 年统计的数据比，增长了 4296 万，互联网的在线人数达到 73.0%。可以说，互联网已经无处不在，融进了人们生活的各个方面，人们利用互联网便捷自己的生活，比如日常的移动支付、网络购物、网上订餐等等[2]。然而事物都是利弊的形式存在的，以网络购物为例，互联网的发展，带动了电商的兴起，增加了就业方式和就业机会。目前，国内也有很多著名的电商平台，例如：淘宝、京东、拼多多等，这些平台每天都会产生大量的用户数据，而这些数据会呈现爆炸式的增长，这样一来必然会产生信息过载的情况[3]，而且大量的数据给其安全性也带来挑战。

科学家和工程师一直在研究信息过载的问题，并致力于解决的一个难题。许许多多的研究信息过载问题的人员，提出了很多具有建设性的建议，其中分类目录、搜索引擎最具代表性[4]。比如分类目录，而雅虎、DMOZ 就是应用了此种解决方案成为著名的互联网公司，分类目录是将一些网站分类，用户需要查找资料时，根据自己查找的类别去对应的网站查找，国内的 Hao123 也是一个分类目录网站[5]。互联网技术的发展，用户的需求也在也在逐渐地变高，网站类别更是层出不穷，分类目录不能覆盖到全部类别的网站，对于一些新颖的或者冷门一点的网站就不会覆盖到，这样可能会导致用户不能全面的得到其想要的信息[6]。同时，对于不同类型的信息要去相对应的不同类型网站上寻找，会严重降低整个工作的效率、浪费用户的时间，因此这种网站越来越不能满足用户的需求。

于是，科研人员提出了另一种解决方案，即搜索引擎[7]。谷歌靠着搜索引擎在互联网领域站稳脚跟，搜索引擎可以让用户通过输入关键词，查找到自己想要的信息[8]。利用搜索引擎查找信息时，但是在现实生活中，用户无法准确描述出信息时，搜索引擎就无法根据关键字查找用户需要的信息，此时，搜索引擎无法

满足用户的需求，从而导致用户的体验感降低。

针对搜索引擎解决的问题，推荐系统由此诞生，之前的方法都是人主动寻找信息，而推荐系统出现之后却是信息主动找人[9]，这不仅给用户提供了便利，提高的用户体验感，推荐系统和搜索引擎都是一种辅助工具，辅助用户在短时间找到有效信息[10]。推荐系统与搜索引擎不一样的地方，在于用户可以不知道明确的信息时，根据用户的过去的行为数据，仍能给用户反馈需要的信息，它将用户的兴趣爱好建模，推荐满足用户兴趣爱好的信息[11]。在用户知道明确的信息的情况，一般都是搜索引擎辅助用户去查找信息，当在用户没有明确目的的情况下，搜索引擎不能反馈有效的信息，而推荐系统能够推荐给用户感兴趣的东西。推荐系统的出现，不仅节约了用户搜索的时间成本、帮助用户在大量的信息资源中迅速寻找出真正需要的信息，而且还解决了搜索引擎只能关键字搜索，不能准确定位用户的真实意图的问题[12]。

推荐系统能够挖掘长尾物品。Chris Anderson 发表的 “The Long Tail”（长尾）就有对此的研究[13]。并于 2006 年出版了有关长尾的书，书中指出，传统的商品销售受到了互联网的冲击，热门商品只占总物品的 20%，而这 20%的物品提供了总销售额的 80%[14]。互联网的发展，在此条件下，电商由于没有货架成本，与传统零售店相比，电商更能销售更多的商品。传统的零售店货架上都是热门商品，因为考虑到货架以及人工成本，而有些不热门商品也提供了比较可观的销售额的。大多数用户的需求是一些热门商品，但是这些商品不能代表全体用户的需求[15]。对于一些用户来说，很难快速从海量信息中得到需要的信息，对于商家来说，自己生产的产品很难得到广大用户的关注。推荐系统就是辅助用户，帮助用户在大量的信息中迅速找到有效信息，另一方面给商家带来利润，实现用户和商家的共赢[16]。

本文以登顶音乐类 App 用户的榜首的网易云音乐为例，网易云音乐最初的目标是建立一个音乐社交网络，网易云音乐侧重于小众圈子，提高了对小众人群的关注，而小众人群有比较大的概率形成社群，经过这种网络关系来提升品牌价值。根据易观分析发布的《2021 中国在线用户洞察报告》中的数据显示，如图 1 所示，相比酷我音乐、酷狗音乐和 QQ 音乐等，网易云音乐是年轻用户占比最大的平台，35 岁以下的用户约占 80%。根据《2020 年网易云音乐销售手册》显示：

在网易云的用户中，以学生及白领、15-35 岁、高学历、一二线城市、可支配收入高的等群体为主，听歌的人群更加年轻化。

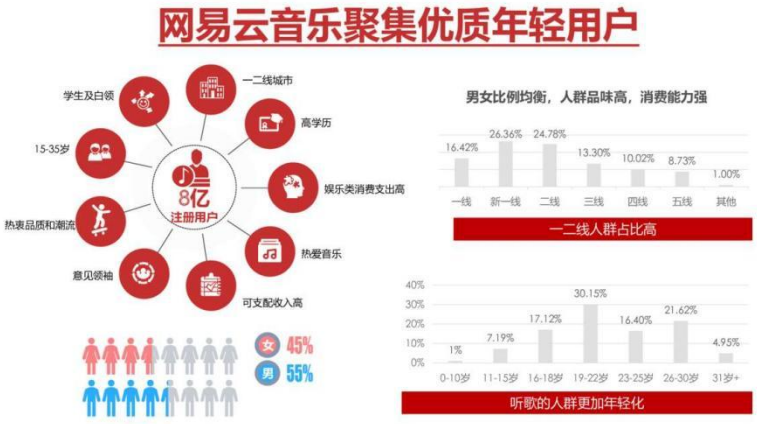

图 1. 2020 年网易云音乐销售手册

Figure 1. 2020 NetEase Cloud Music Sales Manual.

因此可以看出，网易云音乐很受大众欢迎的，因为它能够准确给用户推荐喜欢音乐，推荐系统在网易云的应用使得网易云在音乐领域占据比较重要的地位，有利于提高网易云音乐的核心竞争力，因此推荐系统的应用能够提高行业竞争的竞争力，有利于提高用户的音乐体验，也有利于更加直观地剖析音乐数据，辅助音乐公司做出决策。

1.2 国内外研究现状

上世纪 90 年代初，就有很多学者提出推荐系统这一概念并进行研究，如今推荐系统已成为一门独立的学科，取得了不错的成果，在社交、娱乐、电子商务等领域都得到了应用[17]。比如社交软件 Facebook、Twitter 都应用了推荐系统，电影和视频领域，Netflix 和 YouTube 也引入了推荐系统，当然电子商务领域也有着非常成功的应用，比较著名的电商平台亚马逊，音乐电台 Pandora 和 Last.fm 等等都是推荐系统应用非常成功的例子[18]。

20 世纪 90 年代初，推荐系统概念就提出，一个公司为了解决其研究中心的信息过载问题设计了邮件系统，后来，知名团队明尼苏达大学研究组，开发了第一个自动化推荐系统，后来研究组推出来学术界研究推荐算法的常用数据集 MovieLen。随着推荐系统的不断研究，著名的互联网公司 Yahoo 发现个性化推荐具有潜在商机，因此推出了 MyYahoo。美国的两位学者 Paul Resnick 和 Varian，

在学术界首次提出了推荐系统，标志这推荐系统开始成为一个重要的研究领域[19]。1998 年协同过滤算法应用在亚马逊电商平台，让亚马逊获取了巨大的利润，据统计，20%~30%销售额都是推荐系统带来的，Adomavicius 团队发表论文，将推荐系统分类，并指出未来研究方向[20]。后来，Netflix 举办一场目的为提高电影推荐准确度的竞赛，很大程度上促进了推荐系统的发展，学术界对此次竞赛也表示了极大的关注。后来，美国举行了首届 ACM 推荐系统大会，在大会上，不同领域学者们分享了自己的研究成果，2018 年阿里巴巴的一篇论文被美国人工智能协会录用。

近些年来，国内的很多方面也应用到了推荐系统，国内几大电商平台，京东淘宝等，近几年来短视频的兴起，各大短视频平台也都应用了推荐系统，比如抖音、快手，新闻领域也有推荐系统的身影，比如今日头条，音乐产品如网易云、豆瓣电台极具代表性，关于推荐系统的应用有很多，例如邮件、广告等。

1.2.1 音乐推荐系统研究现状

推荐系统广泛应用于在音乐类的产品中，国际上著名的音乐产品有 Pandora（潘多拉），Last.fm 等，国内的音乐产品也很好，比较有代表性的有豆瓣电台、网易云音乐等，这些著名的音乐平台，都有着自己独特的技术支持。

Pandora 支持的后面技术是基于内容的音乐推荐算法，由专业人员亲自为不同歌手的歌进行标注，主要是对歌曲的特性细化，比如歌曲的编曲、乐器搭配、乐器演奏特征、旋律等维度。然后，Pandora 电台根据专业人员的标注，计算歌曲的之间的相似度，并把用户之前喜欢的音乐相似的音乐推荐给用户[21]。

Last.fm 与 Pandora 的推荐算法不同，它没有使用专家标注，而是记录了用户的行为数据，比如用户听歌次数、用户对歌曲的评分，然后计算出不同用户在歌曲上的喜好相似度，把和相似听歌爱好的用户建立了一个社交网络[22]，这样电台能向用户推荐自己相似爱好的用户听过的歌曲，加强了用户与用户之间的联系[23]。

在推荐系统领域中，音乐推荐是一个比较独特的存在，研究 Pandora 的研究人员，在一次大会上对音乐特点做出了总结[24]：音乐的数量规模很大，种类丰富；用户喜欢听歌曲可能很多、对于喜欢的音乐用户会重复听很多次，用户反复收听这个行为远远高于电影和书，基本上用户读书或者看电影只看一遍，因为用户已

经知道书和电影里面的情节和内容，而且用户收听音乐时会受一些外在或者内在因素影响，例如，新年的时候，用户会收听比较喜庆的歌曲，来衬托节日的气氛，会在伤心难过的时候收听伤感的歌曲，而且用户还可以分享音乐给自己的好友，具有一定的社交性质。

1.2.2 音乐推荐方法研究现状

推荐系统在音乐领域有着非常多的应用，促进了音乐网站的发展，为音乐公司带来可观的利润，但是仍然存在一些比较难以解决的问题[25]，当一个新用户刚刚加入系统的时候，系统没有用户兴趣爱好的任何信息，而用户也不能准确地描述自己喜欢的音乐，此时音乐推荐系统就很难为用户推荐符合用户兴趣爱好的音乐，这时候会因为推荐的不准确性，降低了用户的体验感，给音乐公司带来一定的用户损失[26]。所以音乐推荐算法研究的是如何个性化推荐的问题。用户产生的各种行为数据信息，如何有效利用信息算法进行改进，是音乐推荐系统的重中之重[27]。随着听音乐的用户的增加，音乐市场的需求越来越高，音乐推荐的问题也逐渐出现，包括推荐的响应时间、准确性都需要去解决[28]。音乐推荐系统大多数使用基于协同过滤、基于内容的推荐方法，主要分为以下两类：

（1）基于协同过滤的推荐

协同过滤就是一个利用集体智慧的方法。集体智慧（Collective Intelligence,CI）是一种共享或者群体的智能[29]。在网络还没有出现之前，它就广泛存在于生物学、计算机等领域，随着 web2.0 的迅速崛起，集体智慧在社交网络、推荐等领域也发挥了极大的作用。

在基于协同过滤推荐算法研究中，用户对音乐所做的行为，比如用户搜索歌曲行为，收听歌曲的行为，对音乐打标签和对音乐评论等行为，系统都不仅将这些行为收集，还收集一些比如分享、收藏等重要的隐式反馈[30-31]，该推荐算法会根据用户的行为数据进行推荐。文献[32]提出了一种方法新的相似性度量标准，给更具影响力的用户赋予一定权重进行推荐。Andreu Vall 等提出引入的特征组合混合推荐系统能够更准确地预测拟合的播放列表继续，改善了播放列表中的少数歌曲的长尾[33]。基于协同过滤的音乐推荐系统依旧存在挑战，例如，如何利用现有可用数据向用户推荐、冷启动等，这些仍然是协同过滤需要解决的问题。

（2）基于内容的推荐

基于内容的推荐。用户对物品的行为数据对该算法不产生影响，只需要提取物品本身内容信息[34]。如何从物物品本身信息得到用户的兴趣数据是一个问题，这时需要利用一些用机器学习的方法。该算法也有很多学者深入研究，提出不少代表的建议，而且推荐效果不错。文献[35]分析标签与音乐特征属性，提出了一套针对多标签类型分类任务的集成技术。通过组合多个分类器来提高音乐标签注释系统的性能； Zheng E 等人实现了基于知识标签个性化动态推荐系统的优化[36]。Cyrillic Laurier 等人提出通过分析人们情绪来标记音乐，根据这些数据创建特征，结合聚类的方式，体现出音乐与情感属性的映射关系[37]。

1.3 本文的组织结构

第 1 章 绪论。本章首先介绍推荐系统的研究背景与意义。其次，介绍了国内外研究现状，分析了音乐推荐系统现状。最后介绍了本文了组织结构。

第 2 章 相关理论及技术的介绍。本章介绍得到音乐数据集的爬虫的网络相关技术，工作原理以及分类。然后从原理、以及优缺点对音乐推荐算法进行了详述，因为不管是什么推荐算法，相似度的计算都是最为关键的，这里对其进行了介绍。最后介绍了音乐系统开发技术，主要介绍了前端以及后端涉及到的框架。

第 3 章 基于协同过滤和内容的音乐推荐算法。在协同过滤的基础上改进相似度的计算。针对协同过滤算法中的冷启动问题，提出利用用户的标签对歌曲的热门度进行定义，解决冷启动问题。最后，针对上述两个推荐算法，在单独使用时的不足，将协同过滤和基于内容算法两种算法进行混合。

第 4 章 音乐推荐系统的设计与实现。首先介绍了数据来源以及实验环境。其次，介绍了系统分析，从需求分析以及可行性分析两个方面，对系统进行分析，然后，介绍了系统功能模块的设计。而后，介绍了数据库设计，主要是一些主要实体设计与数据表以及系统界面图的实现。

最后为总语，对本文进行总结与展望。本章对全文所做工作进行汇总分析得出结论，为后续进一步研究音乐推荐系统，改善推荐效果奠定基础。

1.4 本章小结

本章介绍了在互联网技术快速发展的背景下为解决信息过载的问题提出了解决方案：分类目录、搜索引擎及推荐系统。对它们进行逐一分析，以网易云音乐为例，指出音乐对于年轻人的不可或缺性并顺势提出建立音乐推荐系统的必要性，然后对推荐系统、音乐推荐系统和音乐推荐方法的研究现状进行了详细介绍，并简单地介绍了文章的组织结构。

第 2 章 相关理论及技术介绍

2.1 网络爬虫技术

2.1.1 爬虫工作原理

网络爬虫也称为“蜘蛛”，它可以在海量的互联网信息爬取需要的信息。简单地说它是模拟类请求网站的行为，即自动请求网页、抓取数据，然后从中提取有价值的数据[38]。具体步骤如下，首先发送请求获取目标网页，通过分析页面获得网页的源代码。其次，解析页面从网页源代码中提取出本研究所需的数据。该操作为数据的处理以及分析提供便利，因此需要给予高度重视。最后，以适当的格式保存抽取的部分数据。通常以 TXT 文本、CSV 或 JSON 等格式将数据保存在文本中。

2.1.2 爬虫的分类

爬虫技术主要有通用爬虫（搜索引擎的爬虫）和聚焦爬虫（针对特定网站的爬虫），主题式爬虫聚焦爬虫（又称主题式爬虫），是一种有特定主题的爬虫技术。它与通用爬虫原理类似，不同的就是聚焦爬虫会进行过滤与主题无关的 URL，能够帮助用户获取只和目标主题相关的网页内容，提高了爬虫的精确度这里采用的是聚焦爬虫。

（1）通用爬虫

该种类型的爬虫技术没有特定要求，首先根据初始化的 URL，进行爬取此 URL 对应网页的内容和其他 URL，将此过程中得到的新的 URL 未经爬虫，就将其放入未爬 URL 队列中，不断循环往复，直到达到满足的条件停止。

（2）主题式爬虫

主题式爬虫又称聚焦爬虫，是一种有特定主题的爬虫技术。它与通用爬虫原理类似，不同的就是聚焦爬虫会进行过滤与主题无关的 URL，能够帮助用户获取只和目标主题相关的网页内容，提高了爬虫的精确度[39]。

2.2 常见的音乐推荐算法

主要应用于音乐的推荐系统的推荐方法有：协同过滤算法、基于内容的推荐，

以及混合推荐算法。

2.2.1 基于协同过滤的推荐算法

协同过滤算法是应用最广泛。该算法主要依据用户的反馈的一些行为信息，计算用户、物品之间的相似度，找到类似的用户或物品，然后计算预测出目标用户对物品的评分，按照评分的高低形成推荐列表，最后给用户进行推荐[40]。简单地来说，是在海量的用户中发掘出自己的邻居用户，也就是品味类似的用户，然后根据他们喜欢的东西组成一个推荐列表，然后根据推荐给你。

（1）基于用户的协同过滤算法的原理

基于用户的协同过滤（User Based Collaborative Filtering，UserCF）算法参考了“人以群分”的思想，行为喜欢相似的人比较容易成为一个群体，比如当那你想去电影院看电影的时候，总是喜欢询问好友的意见，问最近有什么好看的电影，这一行为本质是将相同特征的用户分成一类。

该算法的核心思想是把不同用户对项目的反馈的数据信息，模拟成向量，计算两个向量之间的相似值，根据相似度，找出同目标用户的相似用户，将相似用户买过的物品，而目标用户没有买过，推荐给目标用户。简单地来说，该算法就是利用一类用户的兴趣信息，为目标用户进行推荐。该推荐算法的过程如图 2 所示。

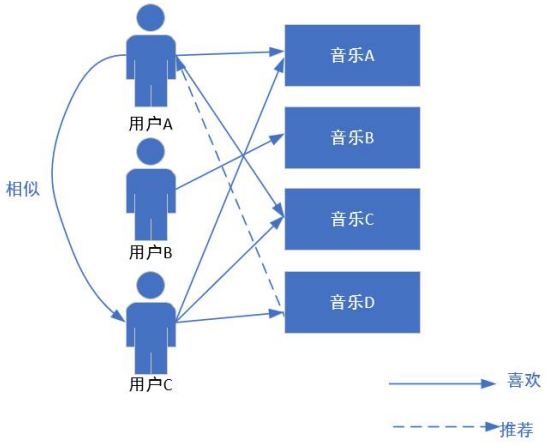

图 2. 基于用户的协同过滤推荐图

Figure 2. User-based collaborative filtering recommendation graph.

用户 A 喜欢音乐 A 和音乐 C，用户 C 喜欢音乐 A、音乐 C 和音乐 D，因此将用户 A 没有表达喜好的音乐 D 推荐 A,这里主要总结为两步：首先计算用户 C

的相似用户，然后找到这些相似用户喜欢的但 C 没有进行过评分的物品并推荐给 C。

（2）基于用户的协同过滤算法的优缺点

随着学者们不断深入研究，也发现了此算法的优点及缺点，总结了推荐算法一些优点及缺点如下：

基于用户的协同过滤算法的优点有：①相似用户给出的评分数据信息能够被此算法有效利用，推荐的效率大大提高。②此算法利用了用户的反馈信息，所以能够发现用户自己也不知道的兴趣的爱好，算法实现起来也比较简单，而且推荐个性化程度高。③对于用户比较少的时候能发挥很好的作用[41]。

缺点也很明显如下：①历史数据的稀疏，数据稀疏问题是此算法至今为止最难的挑战之一，在数据稀疏的情况下，很难计算用户之间相似度，导致在使用协同过滤算法在推荐结果时出现偏差，推荐质量收到影响，降低用户体验感。②冷启动问题：在实际生活中，更极端的境况是，新用户刚刚加入一个系统时，没有历史行为数据，这就是冷启动问题。冷启动研究的时在没有大量数据的情况下，推荐系统给用户推荐所能接受的内容，用户冷启动、物品冷启动以及系统冷启动是冷启动的三个基本类别。

（3）基于物品的协同过滤的原理

基于物品的协同过滤参考了物以类聚的思想，例如用户喜欢的物品具有某一类特征，于是具有某些特征的物品就被分在一类，最为广泛应用在电子商务的领域的，比如去超市买饼干，会发现各种各样的饼干放在一起放在货架上售卖，有曲奇饼干、威化饼干等等。

此算法的基本思路，是根据目标用户感兴趣的物品，计算物品之间的相似度然后为用户做出推荐。首要用一些代表性的特征表示被推荐的产品，其次根据目标用户之前的行为数据，即对物品特征的喜好，喜欢或者不喜欢，分析出这个用户的兴趣偏好，然后将待选物品的特征符合这个目标用户的偏好的物品，形成一个产品集，按照符合程度高低排序。如图 3 所示。

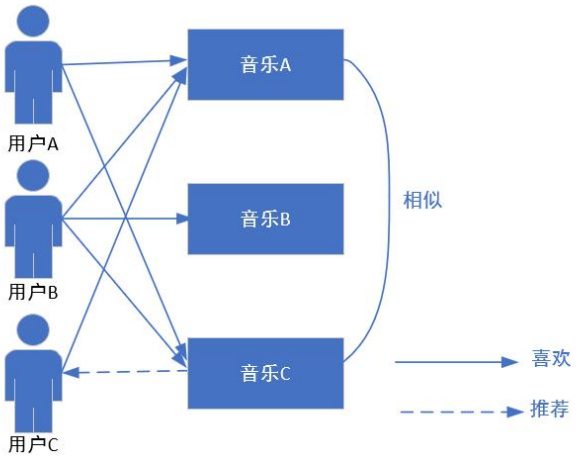

图 3. 基于物品的协同过滤推荐图

Figure 3. Item based collaborative filtering recommendation graph.

用户 A 喜欢音乐 A 和音乐 C，用户 C 喜欢音乐 A，音乐 A 和音乐 C 相似，因此将用户 C 没有表达喜好的音乐 C 推荐给用户 C，这里主要总结为给用户推荐之前喜欢音乐的相似音乐。

（4）基于物品的协同过滤推荐算法的优缺点

基于物品的协同过滤推荐算法的优点描述如下：

①当用户数量远远超过物品数量时，计算相似度时计算量小不需要频繁更新。②用户与用户独立：使用该算法时，用户的行为数据信息是有用的，与其他用户之间的关系此算法不需要考虑，而且用户的评价对使用此算法推荐时不产生的影响，在现实生活中可能商家恶意竞争，故意给物品恶意评价，在这个问题上，该算法比上述基于用户的协同有优势，并解决了它不能解决的用户行为数据稀疏的问题。③避免新物品的冷启动问题，此算法是利用物品的特征，然后进行分析推荐，所以避免了新的被推荐物品的冷启动问题。

此算法的缺点描述如下：

①不能挖掘其他潜在兴趣爱好，该算法没有考虑用户的兴趣信息，只利用了喜欢的物品信息，所以推荐给目标用户的物品都具有某一特征，不能发现目标用户可能存在其他的喜好。②推荐的物品多样性低：因为该算法推荐的物品，都是从之前喜欢的物品中提取出的特征，物品覆盖面比较小，并不能完全代表目标用户喜欢的所有物品，所有丰富度比较低[42]。

2.2.2 基于内容的推荐算法

该算法是最早出现的推荐算法，主要是用于信息检索，算法思想也比较简单，它是根据用户之前喜欢什么样的物品，推算出来用户可能还喜欢什么物品，并且把这些物品推荐给用户。基于内容的推荐是向用户推荐，之前所喜欢物品的相似物品，分为三步：第一步是构造物品的特性；第二步是构建用户偏好矩阵；第三步是计算推荐。这里的“内容”指的是用户之前喜欢的物品，并由此推算出来的用户偏好，如图 1 所示，基于内容推荐的一个简单的例子，用户 A 对治愈放松类型的音乐非常感兴趣，用户 C 也对此类音乐感兴趣，用户 B 对摇滚、说唱类型音乐很感兴趣，用户 A 没有收听音乐 C，就把用户 C 听过的音乐 C 推荐给用户 A。如图 4 所示。

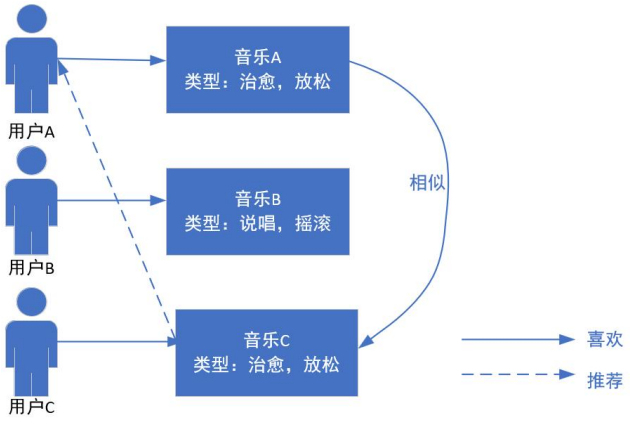

图 4. 基于内容推荐图

Figure 4. Content-based recommendation graph.

基于内容推荐的优缺点如下：

该算法的优点：①能够解决冷启动问题。这点协同过滤是不能解决的；②能够解决数据稀疏问题。协同过滤遇到数据稀疏时，不具有优势。缺点有就是不能发现用户潜在兴趣，推荐的也只是和之前相似的物品，不具多样性。

2.2.3 混合推荐算法

由于之前常用的推荐算法在单一使用的时候，他们各有优缺点，不能完美结局各种场景，基于内容的推荐算法侧重物品本身的属性，这就导致了在个性化程度不够，协同过滤算法侧重用户的历史行为数据，但是它在多样性上不足，一些

学者就提出了结合多种推荐算法产生了混合推荐算法[43]，结合他们的优点解决问题。Netflix Prize 竞赛就是一个不同推荐算法混合在一起的例子，这次竞赛目的是为了改进一个电影领域的推荐系统，提高整体的准确率[44]。混合推荐为分以下三种基本设计思想：

（1）整体式

整体式混合设计，是指很多的推荐算法整合到一个算法中，在事实上是多个推荐算法起作用，而实现的混合设计，如图 5 所示。

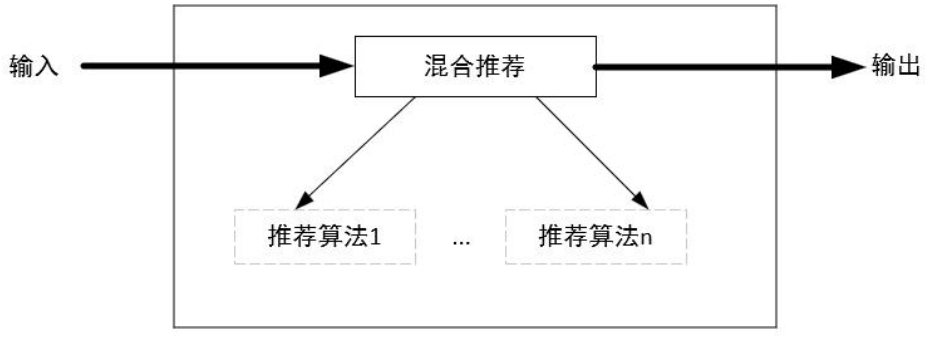

图 5. 整体式混合设计

Figure 5. Integrated hybrid design.

（2）并行式

并行式混合设计，指的是根据输入的不同，使用的推荐算法不同，他们之间相互之间独立，各自产生自己的推荐列表，如图 6 所示，之后产生的结果被组合到最终的推荐集合中。

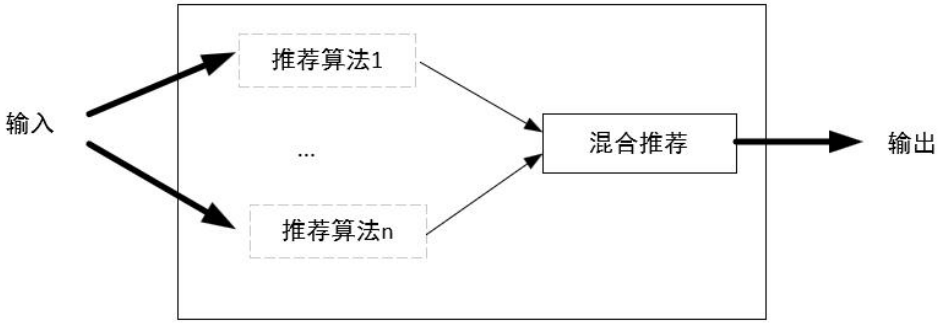

图 6. 并行式混合设计

Figure 6. Parallel hybrid design.

（3）流水线

流水线混合，指的是前一个推荐系统的输出，是后面一个推荐系统的输入部分。如图 7 所示

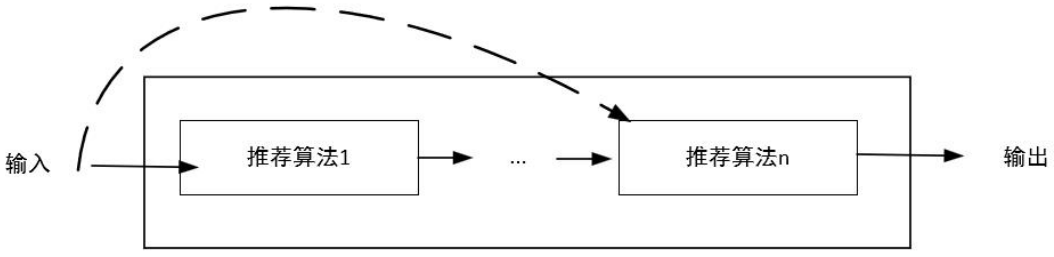

图 7. 流水式混合设计

Figure 7. Flowing hybrid design.

2.3 相似度的计算方法

上一节介绍了几种推荐算法，无论是什么推荐算法，相似度的计算都是最基本的问题。相似度用来衡量物品与物品之间、用户与用户之前是否相似，而它有许多计算方法有很多，接下来介绍相似度计算的几种传统的方法[45]，皮尔逊（Pearson）相关系数，杰卡德（Jaccard）相似度以及余弦相似度。

2.3.1 皮尔逊相关系数

皮尔逊相关系数（Pearson）用来衡量物品、用户之间的线性相关程度，取值范围在 1 和-1 之间。皮尔逊相关系数的计算如式（2-1）所示。

$$\mathrm{Sim}(X,Y)=\frac{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i-\bar{X}\right)^2\times\sum_{i=1}^{n}\left(Y_i-\bar{Y}\right)^2}} \tag{2-1}$$

其中，用户 X 对项目 i 的评分用 X_i 表示，用户 Y 对项目 i 的评分用 Y_i 表示，用户 X 打分的平均值用 $\bar{X}$ 表示, 用户 Y 打分的平均值用 $\bar{Y}$ 表示。

2.3.2 杰卡德相似度

Jaccard 相似度是用来衡量的两个项目之间的相似度。Jaccard 相似度的计算方法，两组集合的交叉元素数除以两组集合的并集元素数，由此得到相似度计算

结果，其中集合元素指的是用户喜欢的项目集合，或这是喜欢该项目的用户集合，计算如式（2-2）所示。

$$\mathrm{Sim}(a,b)=\frac{\left|S_a \cap S_b\right|}{\left|S_a \bigcup S_b\right|} \tag{2-2}$$

公式中，S_a代表的含义是用户 a 喜欢的项目集合，S_b代表的含义数用户 b 喜欢的项目集合，相似度的取值范围在 0 到 1 之间。当两个项目之间没有共同的特征，相似度的值接近 0，表示它们是两个不同样式的项目。相似度的值接近 1，表示两个项目间的相似性很高。

2.3.3 余弦相似度

余弦相似度它表示两个样本数据间的相似度。余弦相似度常在协同过滤推荐算法中，用于计算项目之间的相似度，余弦相似度计算式（2-3）所示。

$$W_{AB}=\frac{|N(A)\cap N(B)|}{\sqrt{|N(A)|*|N(B)|}} \tag{2-3}$$

其中，$N(A)$表示用户 A 曾经有过评分物品集合，$N(B)$为用户 B 曾经有过评分的物品集合。W_{AB}表示用户 A 和用户 B 的余弦相似度。W_{AB}的最大值为 1，此时两个项目之间的相似度很高，W_{AB}的值为 0，就表示两个项目之间不相似。

2.4 系统开发技术

2.4.1 前端相关技术

前端采用了 Vue.js 框架设计，JavaScript 框架基础上一个用于构建用户界面的框架，Vue 的核心库只关注视图层，可以通过简单的 API 实现响应的数据绑定，在开发环境下，Vue 会给开发人员一些警告，来辅助开发人员避开常见的错误与陷阱。

2.4.2 后端相关技术

（1）MVC 框架

为了满足大量用户的使用需求，系统采用了 MVC（Model View Controller）

系统架构模式，即模型－视图－控制器的模式，这种模式是软件设计的典范，使用 MVC 的目的是实现代码分离，从而使同一个程序可以使用不同的表现形式，完成对音乐推荐系统的搭建。MVC 架构模式如图 8 所示。

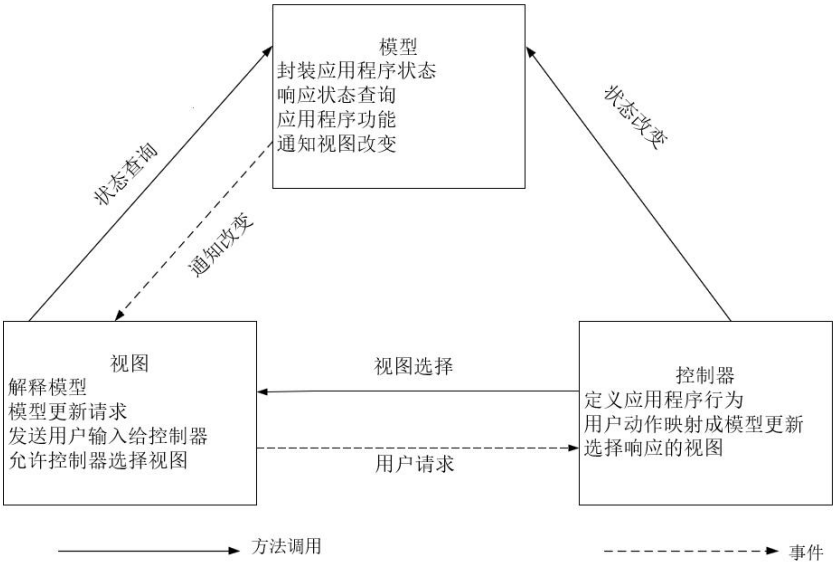

图 8. MVC 体系框架图

Figure 8. MVC architecture diagram.

它把软件系统分为三个模块：①视图，视图代表用户相互界面，MVC 设计模式对视图处理不包括在视图上的业务处理，仅仅是数据采集和处理。视图是提供用户操作页面，可以说是程序的外壳。②模型，模型是业务规则的制定，处理业务流程以及状态，模型接受视图请求，并返回处理结果，此过程对其他层相当于黑箱操作。它负责存储系统核心数据，即程序需要操作的数据以及信息。③控制，控制接受用户请求，将模型与视图匹配在一起，共同完成用户的请求，即根据用户输入的指令，向模型发送数据，负责管理与用户交互控制。

（2）Django 框架

音乐推荐系统后端采用 Django 框架进行开发，Django 是一个开源的 Web 框架，是很多 Web 开发人员最喜欢的框架之一，它是由 Python 写成，是 Python 下一个具有代表性的框架，应用比较广泛，该框架功能强大，集成了 ORM 等多个模块，给开发人员开发项目的时候提供了极大的便利。

2.5 本章小结

本章主要介绍网络爬虫技术，爬虫的工作原理、爬虫的分类，然后几首了几

种常用的推荐方法，基于内容的推荐算法、协同过滤推荐算法以及混合推荐算法。主要介绍的是协同过滤推荐算法，包括基于用户协同过滤、基于物品的协同过滤算法的原理、以及优缺点。最后介绍了推荐系统常用的几种相似度计算方法，以及系统开发的一些相关技术。

第 3 章 基于协同过滤和内容的音乐推荐算法

在传统的协同的过滤算法，存在了一些问题，比如在相似度计算上。热门商品会给计算用户之间、物品之间的相似度时带来影响，降低了推荐系统的推荐质量。实际的情况下，很多用户之间、物品之间并没有交集，这情况下利用余弦计算相似度时，分子为 0，得到的相似度结果为 0，这样的数据没有计算的意义，针对上述相似计算的问题，提出了采用惩罚热门商品和优化算法复杂度来就改进算法，算法流程图如图 9 所示。

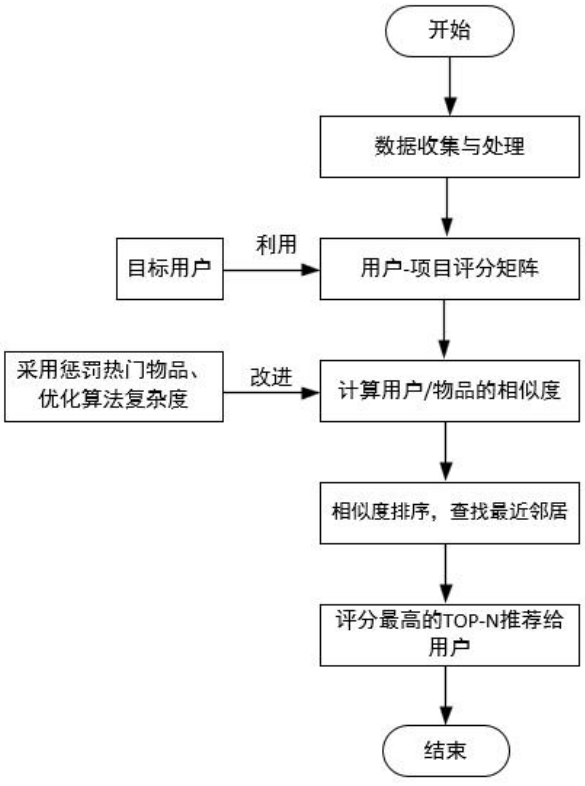

图 9. 协同过滤算法流程图

Figure 9. Flow chart of collaborative filtering algorithm.

3.1 基于用户协同的音乐推荐算法

该算法主要利用用户的之前评分等数据信息，挖掘出用户感兴趣的物品，并且对这些物品进行评分，然后通过同一件物品，不同用户所给的评分，评测用户与用户之间的相似性，给相似用户推荐物品，简单的来说，该算法就是把“和他兴趣相投的其他用户”喜欢的物品推荐给用户。

3.1.1 构建用户-歌曲偏好矩阵

本文把用户对歌曲收听次数等比例转换为到[0，5]区间值作为用户对歌曲的评分，构建用户歌曲表。如表 1 所示，得到用户和歌曲的对应关系。

表 1. 用户对歌曲的评分表

Table 1. Users rating sheet for music.

用户	歌曲 a	歌曲 b	歌曲 c	歌曲 d	歌曲 e
用户 A	3.0	4.0	0	3.5	0
用户 B	4.0	0	4.5	0	3.5
用户 C	0	3.5	0	0	3
用户 D	0	4	0	3.5	3

3.1.2 改进用户相似度的计算

在实际的情况下，很多用户之间并没有交集，也就是并没有对同一首歌曲产生过行为，所以很多情况下分子为 0，使用余弦相似度计算是结果为 0，这样的数据没有计算的必要，传统的协同过滤算法实现将时间浪费在计算这种用户的相似度上，所以只需计算有交集的用户之间的相似度，针对以上优化思路，首先用歌曲到用户的倒排表来该歌曲有哪些用户表达了自己的喜好，得到用户评分过的歌曲，并构建歌曲-用户的倒排表，如图 10 所示。其次根据建立的倒查表,建立用户相似度矩阵 W。

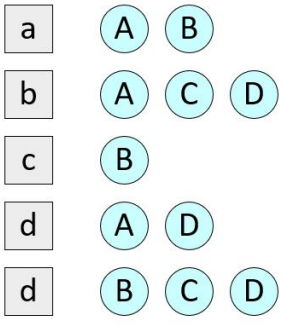

图 10. 歌曲用户倒排表图

Figure 10. Singer user inversion list diagram.

建立用户相似度矩阵 W，计算用户之间的相似度，如图 11 所示

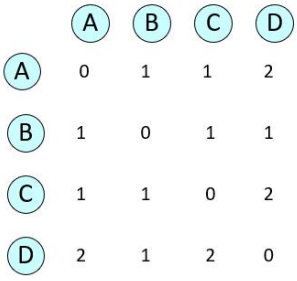

图 11. 用户相似度矩阵图

Figure 11. User similarity matrix diagram.

根据上述相似度矩阵 W，矩阵里的元素，就是计算相似度时的分子部分，然后除以余弦相似度分母部分（根据公式 2-3 得到余弦相似度分母部分）得到用户之间的相似度，以 C 用户为例，从相似度矩阵可知用户 C 与用户 A 相似度计算的分子为 1，与用户 B 相似度计算的分子也为 1，与用户 D 用户相似度计算分子部分为 2，则 C 用户与其他用户相似度计算如下：$W_{CA}=1/\sqrt{6}$ ，$W_{CA}=1/\sqrt{6}$ ，$W_{CA}=2/\sqrt{6}$ 。

在使用该推荐算法给用户推荐喜欢的歌曲的时，如果某一首歌曲非常热门，很多用户都听了这首歌曲，并且对歌曲进行了评分，那么在计算用户之间的相似度的时候，会出现偏差，导致任何两个用户之间的相似度都偏大，从而推荐质量下降，所以本文采用了惩戒热门歌曲的方法，来降低热门歌曲对用户相似度的影响。

采用惩罚热门商品是指，给热度很高的商品会给计算相似度带来一定的影响。如果两个用户都买过字典，这并不能证明两个用户相似，因为字典是工具书，每个人在学习汉字的时候都能用得上，是属于很热门的商品，但是如果两个用户都买个《Python 从门到精通》这本书，那可以大致认为这两个人的兴趣爱好是相似的，因为只有学习编程语言的人才会买着本书。

本文利用公式（3-1），降低热门歌曲的相似度

$$w_{AB}=\frac{\sum_{i\in N(A)\cap N(B)}\frac{1}{\lg(1+|N(i)|)}}{\sqrt{|N(A)||N(B)|}} \tag{3-1}$$

公式中，用 $\lg(1+|N(i)|)$ 代表的含义是对用户 A 和用户 B 共同歌曲列表的中的热度比较高的歌曲进行惩罚，降低了热度比较高的歌曲对计算用户相似度时产

生的影响， $N(i)$表示是对物品有过行为的用户合集，歌曲 i 越热门， $N(i)$越大。

3.1.3 预测歌曲评分并推荐

根据上述步骤算得用户的相似度，然后可以计算出用户对其他未评分的歌曲的评分，采用公式（3-2）计算，公式中， $P_{ucf}(A,i)$代表的含义是用户 A 对歌曲 i 感兴趣程度，$S(A,K)$代表的含义是和用户 A 兴趣相似的 K 个用户，N(i) 代表的含义是收听过歌曲 i 的所有用户，w_{AB}表示代表的含义用户 A 和用户 B 的兴趣相似度， r_{Bi}代表的含义是用户 B 对歌曲 i 的兴趣，也就是用户对歌曲的评分。

$$P_{ucf}(A,i)=\sum_{B\in S(A,K)\cap N(i)} w_{AB}r_{Bi} \tag{3-2}$$

由此可以计算用户 C 对歌曲 a、c、d 的偏好程度，如下：

表 2. 计算推荐结果

Table 2. Calculate the recommended results.

	歌曲 a	歌曲 c	歌曲 d
未改进的评分	2.858	1.837	4.287
改进后的评分	2.061	1.325	3.092

观察上表，不同相似度计算方法的结果有区别，由惩罚了热门歌曲的相似度计算出的用户 C 对为评分的物品的偏好程度比较低，在实际的计算中两者的结果还是有明显的差距的。

3.2 基于物品协同的音乐推荐算法

3.2.1 构建用户-歌手偏好矩阵

构建用户-歌手偏好矩阵。把用户听过的歌曲列表中属于某个歌手的曲目数进行统计，作为用户对于歌手的评分。例如：用户 u_l 的歌曲列表中属于歌手 A 的歌有 5 首，则用户对于歌曲的评分即为 5。经过数据预处理，把数据转化到[0，5]，如表 3 所示：